

SUPEREA

Anno 2
n. 7 - Lug./Ago. 1985

L. 7.500

Spedizione in
abbonamento
postale Gruppo III/70

7

SUPERSINC
la rivista per gli utenti
dei prodotti
SINCLAIR

sinc

Una pubblicazione della

 **J. soft** EDITRICE

in collaborazione con

GRUPPO
EDITORIALE
JACKSON



**UNDICI:
UN SOLITARIO**

**QL: LO STATO
DELLE COSE E...
UN GIOCO**

**CONVERSIONI
PER ZX81**

**LA MAPPA
DI SABRE WULF**

**FOR LOVE:
UNA GRANDE AVVENTURA
PER SPECTRUM 48K**

Ritorna in edicola

VIDEO BASIC

Il corso più entusiasmante su cassetta
del Gruppo Editoriale Jackson per Commodore 64,
VIC 20 e Spectrum

200.000 copie vendute

del 1° fascicolo della prima edizione

Ogni lezione
uno spettacolo

Col 1° fascicolo
una cassetta giochi



Il corso è composto da:
20 fascicoli + (Quattordicinali)
20 cassette +
5 splendidi raccoglitori

Oggi è davvero facile imparare il Basic. Con Video Basic il corso su cassetta che ti permette di programmare subito il tuo computer. È facile: tu chiedi, lui risponde, tu impari. Passo dopo passo. Sul tuo schermo appaiono le domande, le risposte, gli esercizi e

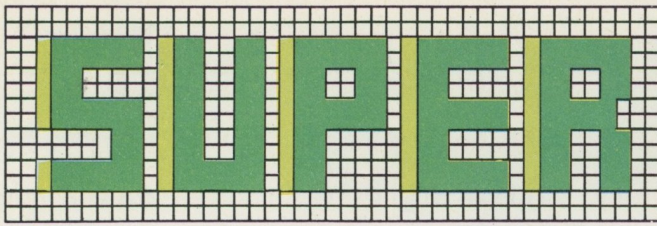
tu, senza fatica, presto e bene, impari a conoscere e programmare il tuo computer, sia esso un VIC 20, un Commodore 64 o un Sinclair. Video Basic è in edicola. Provalo subito. Ogni lezione è uno spettacolo.

Oggi il Basic si impara così. Video Basic, il corso su cassetta per parlare subito col tuo computer.

Video Basic
per imparare non solo il Basic.



Un'altra grande idea firmata
GRUPPO EDITORIALE JACKSON
Milano-San Francisco-Londra-Madrid



5 EDITORIALE
POINT

DOVE VA L'HOME COMPUTER?
a cura di **Pietro Dell'Orco**

7 **READ & WRITE**
LA POSTA DEI LETTORI

11 **OPEN**

QL, LO STATO DELLE COSE
di **Lucio Bragagnolo**

18 LA CULTURA **OS** INFORMATICA
...then

FORTH: IL LINGUAGGIO DELLA QUARTA GENERAZIONE
di **Marcello Spero**

23 **REM:HW**

MISCELLANEA
di **Marcello Spero**

29 **RANDOMIZE**

L'ASSEMBLY ASSIEME
di **Marcello Spero**
Quarta parte

37 **RANDOMIZE**

SABRE WOLF: UNA GIUNGLA 16x16
di **A. Barbati e F. Bruschi**

40 **LOAD**

40 CANCELLINO
di **S. Rodgers**
trad. e adatt.
di **L. Brambilla**

41 COLSPOT
di **E.E. Minetti**

45 CONVERSIONI BASI NUMERICHE
di **G. Bertolotti**

48 SUPERCOMPOSTER
di **J. W. Wassovi**
trad. e adatt.
di **Paolo Maffei**

51 UNDICI
di **C. Gooch**
trad. e adatt.
di **A. Violini**

56 STAR WAR
di **L. Lotti**

60 LOG HOP
di **C. Gibbs**
trad. e adatt.
di **A. Violini**

63 MEMOGAME
di **S. Ghisoni**

67 FOR LOVE
di **Mark Chapman**
trad. e adatt.
di **Carlo Panzalis**

69 RIPETIZIONE DEI TASTI
di **Paolo Maffei**

81 TOMBOLA 1K
di **Guido Bertolotti**

83 SPECTRUM RENUMBER
di **Marco Broglia e Davide Conti**
Seconda parte

92 **PRINT**

RECENSIONI LIBRI
di **Lucio Bragagnolo**

94 **ZX CLUB**

I CLUB SINCLAIR

J. soft s.r.l.
DIREZIONE, REDAZIONE, AMMINISTRAZIONE
Viale Restelli, 5
20124 MILANO
Tel. (02) 68.88.228-68.37.97
68.80.841/2/3

DIRETTORE RESPONSABILE:
Pietro Dell'Orco

COORDINAMENTO TECNICO:
Riccardo Paolillo

REDAZIONE:
Lucio Bragagnolo
Marcello Spero

HANNO COLLABORATO A QUESTO NUMERO
A. Barbati - G. Bertolotti
L. Brambilla - M. Broglia
F. Bruchi - D. Conti - S. Ghisoni
L. Lotti - E.E. Minetti
C. Panzalis - A. Violini

GRAFICA E IMPAGINAZIONE:
Margherita La Noce
Ivana Rossi
Raffaella Toffolatti

FOTOCOPOSIZIONE:
Graphotek - Via Astesani, 16
Tel. 02/64.80.397
20161 MILANO

CONTABILITÀ:
Giulia Pedrazzini
Flavia Bonaiti

AUTORIZZAZIONE ALLA PUBBLICAZIONE:
Tribunale di Milano n° 199
del 14.04.1984

STAMPA:
Litografia del Sole
Albairate (MI)

PUBBLICITÀ
Concessionaria per l'Italia e l'Estero J. Advertising s.r.l.
Viale Restelli, 5
20124 Milano
Tel. (02) 68.82.895-68.80.606-68.87.233
Tlx. 316213 REINA I

Concessionaria esclusiva per la DIFFUSIONE in Italia e Estero:
SODIP - Via Zuretti, 25
20125 MILANO

Spedizione in abbonamento postale Gruppo III/70
Prezzo della rivista L. 7.500
Numero arretrato L. 15.000
Abbonamento annuo: L. 82.500
(11 numeri con cassetta) L. 38.500
(11 numeri senza cassetta) per l'estero: L. 110.000 (11 numeri con cassetta) L. 50.000 (11 numeri senza cassetta)
i versamenti devono essere indirizzati a:
J. soft s.r.l.
Viale Restelli, 5
20124 MILANO

mediante emissione di assegno bancario, vaglia o utilizzando il c/c postale n. 19445204.

Per i cambi di indirizzo indicare, oltre al nuovo, anche l'indirizzo precedente ed allegare alla comunicazione l'importo di L. 500, anche in francobolli
© TUTTI I DIRITTI DI RIPRODUZIONE O TRADUZIONE DEGLI ARTICOLI PUBBLICATI SONO RISERVATI

GLI ARTICOLI TRADOTTI SONO TRATTI DALLA RIVISTA ZX COMPUTING - ARGUS SPECIALIST PUBLICATION LTD 1 GOLDEN SQUARE LONDON W 1R3AB



Rivista associata all'Unione Stampa Periodica Italiana

Per collaborare a SUPERSINC

La maggior parte dei seguenti suggerimenti ha lo scopo di migliorare l'accuratezza e la velocità di pubblicazione di un articolo; rispettando questi consigli si accresceranno le probabilità che un vostro lavoro venga pubblicato. La rivista è interessata ad articoli e programmi riguardanti la linea Sinclair. Siamo più interessati al contenuto di un articolo piuttosto che al suo stile, e soprattutto gli articoli devono essere chiari ed esaurienti. La seguente guida permetterà che le vostre buone idee e i vostri programmi vengano più facilmente accettati per la pubblicazione:

1 l'angolo superiore sinistro della prima pagina dovrà contenere: nome, cognome, indirizzo, numero telefonico, codice fiscale, data di spedizione, luogo e data di nascita.

2 l'angolo superiore destro della prima pagina dovrà contenere la marca e il tipo di computer al quale il lavoro si riferisce, unitamente alla configurazione richiesta (memoria occorrente, eventuali periferiche e così via).

3 il titolo sottolineato dell'articolo dovrà iniziare a due terzi in altezza della prima pagina.

4 le pagine seguenti potranno essere battute normalmente, con la condizione che l'angolo superiore destro contenga un'abbreviazione del titolo e del cognome, unitamente al numero di pagina progressivo. Per esempio, Horace Goes.../Brambilla/2.

5 tutte le linee del testo dovranno essere battute con spazio 2 o spazio 3, e un margine di circa un centimetro dovrà trovarsi ad entrambi i lati dello scritto.

6 dovrà essere usata una carta formato A4 e lo scritto dovrà occupare un solo lato del foglio (caratteri maiuscoli e minuscoli).

7 i fogli dovranno essere uniti con una clip.

8 avendo intenzione di spedire più di un articolo, questi dovranno essere inviati separatamente insieme alla rispettiva copia su supporto magnetico.

9 programmi brevi (meno di 20 linee) potranno essere inseriti nel testo, mentre programmi più lunghi dovranno essere listati separatamente. È **ESSENZIALE** per noi disporre di una copia del programma registrata più volte su supporto magnetico, su entrambi i lati dello stesso. È preferibile usare nastri di buona qualità e di lunghezza non eccessiva; la cassetta o il la cartuccia per Microdrive dovranno essere etichettati con il nome dell'autore, il titolo dell'articolo, il titolo del-

l'articolo, il computer interessato e soprattutto le eventuali espansioni richieste. Come suggerimenti di programmazione si consiglia di usare per esigenze di stampa listati, le istruzioni INK, PAPER, INVERSE piuttosto che scrivere direttamente in INVERSE VIDEO. Un rapido controllo dei programmi per operare queste sostituzioni sarà da noi estremamente apprezzato.

10 per maggior chiarezza, all'interno dell'articolo è conveniente usare caratteri maiuscoli riferendosi a istruzioni BASIC (esempio RETURN, LIST, RND, PRINT etc.). Se si desidera evidenziare una parola, è preferibile sottolinearla piuttosto che scriverla in carattere maiuscolo.

11 gli articoli ed i programmi potranno avere qualsiasi lunghezza - da una routine di una sola linea fino a programmi molto complessi.

12 volendo includere fotografie, questi dovranno avere formato 24x36, o 6x6, in bianco e nero o diapositive.

13 non prenderemo in considerazione articoli che siano stati sottoposti ad altre case editrici.

14 il compenso per la collaborazione prestata sarà commisurato alla complessità e all'interesse del programma (da un minimo di L. 50.000 a un massimo di L. 300.000). Il pagamento è effettuato in caso di pubblicazione del lavoro.

15 il materiale ricevuto e non pubblicato non verrà restituito.

Spedite i vostri lavori a:

SUPERSINC
Via Rosellini, 12
20124 Milano

e saremo lietissimi di pubblicare i contributi migliori.

La Redazione

POINT

**Dove va
l'home computer?**

Giungono dagli Stati Uniti voci allarmistiche circa il presente e il futuro degli home computer e dei piccoli personal computer.

Ci sono fabbriche che hanno dovuto cessare temporaneamente la produzione delle macchine per smaltire le scorte di magazzino.

Forti le critiche mosse a proposito del modo di pubblicizzare la presunta indispensabilità del computer nella casa oltre alle motivazioni, a volte senza troppo senso, con cui tutte le case produttrici hanno cercato di imporre a chiunque l'attrezzo tecnologico.

Da sempre l'Italia si è mossa, ovviamente con tempi diversi, sul modello americano; si sono infatti riscontrate le stesse tendenze del mercato statunitense circa la diffusione, l'utilizzo e quant'altro attiene al mondo del personal computing.

Ora dunque, in linea teorica, anche da noi c'è da attendersi una recessione che potrebbe contrarre la domanda e l'offerta di questi prodotti.

Ad una attenta analisi però si possono riscontrare sostanziali differenze di diffusione, facendo comunque le debite proporzioni, tra il mercato italiano e quello americano.

Sono svariati gli esempi che possono essere citati a conferma della ancora scarsa diffusione del personal nel nostro paese.

Un esempio per tutti potrebbe essere rappresentato dalla situazione nelle scuole dove, nonostante le pressioni esercitate da più parti il calcolatore non è ancora stato adottato massicciamente quale indispensabile ausilio didattico.

Pare che da noi, oltre agli impieghi in ambiti ben precisi, il calcolatore non abbia ancora trovato una precisa collocazione, del resto una delle principali caratteristiche risiede proprio nella possibilità di trasformarsi per assumere ruoli spesso molto diversi tra loro. Ogni utilizzatore deve cercare il migliore modo di sfruttare il proprio "attrezzo" ponendosi sempre precisi obiettivi da raggiungere.

Non ci stancheremo mai di ripetere che non bisogna subire passivamente il calcolatore dandogli da elaborare ogni cosa che capita di trovare pubblicata, ma usare quel materiale come spunto creativo per elaborare nuove realizzazioni.

Siamo convinti che siano ancora moltissimi gli utenti di "prima mano" o persone che aspirano al possesso del personal computer, quindi ancora tanto lavoro da fare anche da parte nostra.

Dunque, per la realtà del nostro paese, la recessione americana non ci deve preoccupare più di tanto; da noi c'è ancora tanto da fare e, forse, gli home e i personal computer non hanno ancora conosciuto la loro alba migliore.

Per finire tutta la redazione augura ai lettori buone vacanze, raccomandando a tutti di non lasciare a casa, o in disparte troppo a lungo, il fedele "amico".

Pietro Dell'Orco

Piccola guida per l'input dei programmi

I programmi pubblicati da SUPERSINC sono stati accuratamente provati e verificati: in questo modo speriamo di avere ridotto al minimo la possibilità di errori. Nel caso comunque un programma dovesse presentare malfunzionamenti, pubblicheremo le modifiche da approntare in uno dei numeri immediatamente successivi della rivista.

Per utilizzare i programmi pubblicati è sufficiente digitare i relativi listati sulla tastiera del calcolatore.

Nel caso dello ZX SPECTRUM, abbiamo previsto una semplice decodifica dei tasti da digitare nel modo grafico, per semplificare la comprensione dei listati.

Come è noto, lo SPECTRUM è provvisto di 2 serie di tasti grafici: una prima serie di 16 caratteri grafici predefiniti (i tasti numerici da 1 a 8 e gli stessi "shiftati") e una serie di caratteri definibili dall'utente nell'ambito di un programma (le lettere da A a U).

In entrambi i casi per ottenere i caratteri desiderati occorre entrare in modo grafico (cursore contrassegnato dalla lettera G lampeggiante), premendo contemporaneamente i tasti CAPS-SHIFT e 9.

Nei nostri listati i caratteri grafici predefiniti sono indicati da una Q seguita dal numero corrispondente al testo che occorre digitare, il tutto racchiuso tra due parentesi graffe.

Ad esempio {G4} significa che occorre digitare il tasto 4, con il cursore in modo grafico.

Analogamente la codifica SG, seguita da un numero da 1 a 8, significa che occorre digitare il relativo tasto numerico, premendo contemporaneamente il tasto CAPS-SHIFT.

Ad esempio quando, si trova la codifica {G2} occorre premere il tasto 2 contemporaneamente al tasto CAPS-SHIFT, ovviamente con il cursore in modo grafico. In entrambi i casi precedenti, quando un simbolo grafico deve essere digitato più volte, i caratteri G o SG della codifica sono preceduti da un numero che specifica quante volte va premuto il tasto grafico indicato.

Così ad esempio {8G5} significa che il tasto grafico 5 va digitato 8 volte e analogamente {4SG1} indica che il tasto grafico 1 e CAPS-SHIFT devono essere battuti 4 volte.

I caratteri grafici definibili (le lettere da A a U in modo grafico) hanno una codifica semplificata: la lettera corrispondente, sottolineata.

Quando in un listato viene presentata, ad esempio, una A sottolineata occorre entrare in modo grafico (al solito premendo contemporaneamente i tasti CAPS-SHIFT e 9) e quindi digitare semplicemente il tasto che contrassegna la lettera A.

Quando leggete	Premete	Vedrete
{G1}		
{G2}		
{G3}		
{G4}		
{G5}		
{G6}		
{G7}		
{G8}		
{SG1}		
{SG2}		
{SG3}		
{SG4}		
{SG5}		
{SG6}		
{SG7}		
{SG8}		

Se non siete già in modo G, entrateci schiacciando contemporaneamente CAPS SHIFT e 9

Se dovete uscire dal modo G, schiacciate 9

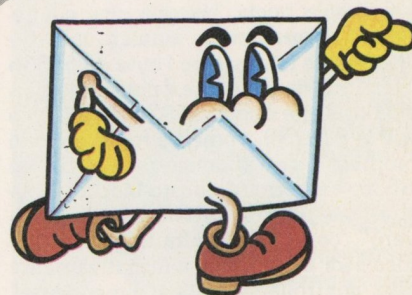
Quando leggete	Premete	Vedrete
<u>A</u>		Simbolo grafico definito nel programma in uso.
<u>B</u>		
<u>C</u>		
<u>D</u>		
<u>E</u>		
<u>F</u>		
<u>G</u>		
<u>H</u>		
<u>I</u>		
<u>J</u>		
<u>K</u>		
<u>L</u>		
<u>M</u>		
<u>N</u>		
<u>O</u>		
<u>P</u>		
<u>Q</u>		
<u>R</u>		
<u>S</u>		
<u>T</u>		
<u>U</u>		

Se non siete già in modo G, entrateci schiacciando contemporaneamente CAPS SHIFT e 9

Se dovete uscire dal modo G, schiacciate 9

READ & WRITE

La posta
dei lettori



Joystick o Kempston?

Possedendo un'interfaccia per joystick programmabile, è possibile simulare l'interfaccia Kempston? Ovvero, avendo un programma che funziona solo con joystick Kempston, posso sostituire quest'ultimo con un joystick programmabile?

Fulvio Gabotti - Cremona

Certo, purché il software di programmazione del joystick non risieda nella RAM dello Spectrum (nel caso in cui si tratti di programmi commerciali).

Assembler 1541

Carissima redazione di Supersinc, sono un felicissimo possessore di uno ZX Spectrum 48K e vorrei sapere se avete intenzione di pubblicare degli articoli sull'Assembler dello Spectrum. Inoltre, mi piacerebbe sapere se lo Spectrum può essere collegato con un floppy disk Commodore 1541.

Andrea Mele

Fizzonasco di Pieve Emanuele (MI)

Carissimo Andrea, la serie di articoli sull'Assembler Z80 sta procedendo già da diversi numeri. Quanto al collegamento con i drive 1541 Commodore si presentano difficoltà tali da sconsigliare un tentativo in questo senso. Tra l'altro, sono ormai reperibili un po' in tutta Italia sia floppy disk drive pensati per lo Spectrum (e quindi interfacciabili senza problemi) sia i cari vecchi microdrive, che nonostante tutte le critiche - eccessive e ingiuste - a loro rivolte sanno svolgere, se trattati con la dovuta attenzione, il loro compito più che egregiamente.

Non si può...

Io vorrei solo una cosa. Potreste pubblicare una cassetta di Atic Atac per ZX Spectrum+, o farmela pervenire?

Mauro Matiddi - Piombino (LI)

I casi sono due: o sei convinto che le cassette per lo Spectrum siano incompatibili con lo Spectrum+ - e per rassi-

curarti basta dire che questo è completamente falso - o (più probabile) ci stai chiedendo di fare copie "pirata" di Atic Atac per poi venderle insieme a SUPERSINC: non abbiamo tuttavia il diritto di duplicare il grande gioco della Ultimate, né sentiamo in noi la vocazione all'appropriazione indebita del lavoro di altri - peraltro già diffusa nel settore informatico ben più di quanto ce ne sia effettivo bisogno.

Tutte le strade portano alla ROM

Potreste pubblicare gli indirizzi delle varie routine della ROM...

Matteo Tosato

Noventa Vicentina (VI)

Come si suol dire, le routine della ROM sono già in pista, e non mancheranno di apparire su SUPERSINC!

A domanda, rispondiamo

Spettabilissima (addirittura! N.d.R.) redazione,

sono anch'io un felice possessore di Spectrum 48K. Desidererei innanzitutto farvi i complimenti per la vostra super rivista e poi porvi tre domande:

1) mi serve una linea di programma che disattivi il BREAK e lo STOP; me la potete fornire?

2) cosa intendete quando dite che gli 80K dell'espansione non sono contemporaneamente disponibili? Cosa comporta questo in un programma?

3) come si fa a sapere quanta memoria rimane nel calcolatore dopo aver scritto un programma?

Ringraziando anticipatamente, vorrei suggerirvi di provare, oltre ai game, anche alcune utility.

Cordiali saluti.

Matteo Rossi

S. Donato Milanese (MI)

Dunque:

1) non esiste un modo realmente efficace di proteggere un programma dal

BREAK rimanendo in ambiente BASIC, in quanto lo Spectrum è sprovvisto dell'istruzione ON ERROR GOTO. L'unico risultato ottenibile con buona facilità è quello di far "crashare" il programma. Ci sono diversi modi per arrivare a ciò; ad esempio, si può inserire una linea tipo IF (condizione indesiderata) THEN POKE 23659,0.

Questa POKE rende impossibile qualunque operazione successiva, tanto che occorre spegnere e riaccendere il computer. Un'altra istruzione micidiale è POKE 23613,0, che ha il vantaggio di evitare il fastidio di dover... spegnere il computer, dato che in caso di BREAK avviene un reset totale. Lo statement più bizzarro è POKE 23613,PEEK(23730)-5: se nel programma viene usato sempre INPUT LINE al posto di INPUT, esso agisce anche contro qualsiasi genere di STOP, compreso CAPS SHIFT+6, con risultati a volte curiosi. Il ricorso al linguaggio macchina è invece inevitabile, se si vuole, in caso di BREAK, mantenere ugualmente in funzione il programma: bisogna realizzare una routine di simulazione di ON ERROR GOTO, che provochi il salto dell'interprete BASIC a una linea del programma stesso, anziché alla ROM dello Spectrum.

2) Cercheremo di essere chiari, anche a scapito della correttezza formale. Lo Z80, come il 6502 Apple/Commodore, come il 6809 Atari, come tutti i microprocessori cosiddetti a 8 bit, può memorizzare internamente solo numeri compresi tra 0 e 65535. In altri termini, lo Z80 può controllare i byte di memoria esistenti all'interno di un computer, partendo dal byte numero 0 per arrivare al byte numero 65535. 80K di memoria equivalgono a 81920 byte: troppi per essere controllati tutti dal microprocessore...contemporaneamente. In effetti lo Z80 può controllare i 16K supplementari solo "dimenticandosi" momentaneamente di altri 16K, in modo che i

READ &WRITE

byte sotto il suo controllo siano sempre non più di 65535. Il problema sta tutto nel riuscire a ingannare il computer con una piccola bugia: bisogna cioè congegnare il programma in maniera tale che sia possibile far passare i byte da 65536 a 81919 (ricordarsi che il conteggio parte da 0, e che quindi 81920 byte vanno da 0 a 81919!), come se fossero quelli compresi tra (poniamo) 49152 e 65535, e alternare questo blocco di byte, all'insaputa della CPU, con quello "vero", così che di volta in volta i due blocchi di byte siano alternativamente sotto controllo, uno, e "dimenticato" l'altro. Per lo Z80 si tratterà sempre di maneggiare costantemente il blocco di byte compreso tra 49152 e 65535, ignorando che si tratta in realtà di due blocchi, uno autentico e l'altro "travestito", che quando ne abbiamo bisogno si scambiano la parte. Tutto questo è realizzabile via BASIC, utilizzando l'istruzione OUT. Torneremo comunque sull'argomento.

3) Questo è molto meno complicato da spiegare: è sufficiente digitare PRINT 65536-USR 7962, ovviamente seguito da ENTER. Lo Spectrum visualizzerà il numero di byte ancora liberi.

Per concludere, ha visto il nostro OPEN dedicato ai migliori Assembler per lo Spectrum, sullo scorso numero? Vedrà che si parlerà ancora di programmi di utilità, anche se non necessariamente in questa veste. Nel frattempo, stiamo cogliendo l'occasione per indicare, in IF THEN, i migliori interpreti/compiler Pascal, LIST e via dicendo...

Stranezze

Caro SUPERSINC, voglio prima di tutto congratularmi per l'ottima idea di pubblicare una rivista riguardante i prodotti Sinclair.

Vorrei sapere se è vero che alcuni programmi per ZX inespanso, salvati su nastro da uno Spectrum 48K, quando vengono caricati in uno Spectrum 16K provocano un reset al termine del caricamento.

Vorrei inoltre che mi venisse spiegata la differenza esistente tra Issue 2 e ISSUE 3 e perché scrivete sulla copertina della rivista i titoli di programmi (come "Galattoidi") che poi non pubblicate.

Riccardo Alezio Bortone

Lecce

Caro Riccardo, il problema da te citato esiste effettivamente, e si verifica su alcuni Spectrum (fortunatamente pochi) che presentano

lievi modifiche hardware rispetto agli altri. Bisogna dire che solo alcuni programmi provocano questo inconveniente, per il quale non esiste rimedio che non sia l'espansione a 48K dello Spectrum in questione.

La differenza tra Issue 2 e Issue 3 è stata spiegata più volte: mentre le IN di lettura tastiera di uno Spectrum Issue 2 hanno come valore di default 255, quelle di uno Spectrum Issue 3 sono a 191; la differenza (64) è sempre la stessa anche per tutti gli altri valori.

Quanto a "Galattoidi" si è trattato di un inconveniente dovuto al fatto che la copertina viene realizzata con grande anticipo rispetto alla rivista, così che non è possibile modificarla in caso di problemi "dell'ultima ora". Va da sé che faremo il possibile affinché la cosa non si ripeta.

L'immortalità dell'anima (di Miner Willy)

Spettabile redazione,

Visto che non avete pubblicato nessun errata corregge riguardo al metodo da voi pubblicato su SUPERSINC del lontanissimo settembre 1984 per ottenere 32 omini in Manic Miner (sistema che non funziona), vi dò addirittura la POKE che fornisce al simpatico Willy nientemeno che l'immortalità.

- Caricate il primo blocco con MERGE"";

- la linea da inserire è 25 POKE 35136,0;

- date RUN e caricate il resto del programma.

Gabriele Barrera - Torino

Il sistema funziona: è solamente stato spiegato in modo probabilmente un po' ambiguo. La POKE dell'immortalità è stata pubblicata da tempo su PAPER-SOFT, ma è passato abbastanza tempo perché valga la pena di ripeterla.

Tastiere, stampanti e guerre simulate

Egregi signori,

complimentandomi innanzitutto per la rivista, vi scrivo per porvi alcuni quesiti:

1) cosa ne pensate della stampante Epson FX-80? Ho occasione di essere a Londra ogni cinque settimane e, vistala sul posto, me ne sono innamorato. Esiste però un centro di assistenza tecnica anche in Italia? E, tecnicamente, qual è il vostro parere? Preciso che me ne servirei per scrivere testi e dovrebbe lavorare parecchio.

2) Quanto può valere la pena di sostituire l'attuale tastiera del mio Spectrum (inten-

do quella coi tasti di gomma) con una Keyboard alternativa? È una sostituzione che si può operare autonomamente oppure è necessario l'intervento di un tecnico?

3) Tra i vari game che ho sperimentato ho preferito in assoluto simulazioni strategiche del tipo "Battle of Britain", "War 70" e "Stonkers". Cosa esiste in commercio di simile (e possibilmente migliore)?

4) Esiste in commercio per lo Spectrum un programma che esprima graficamente, mediante istogrammi e diagrammi, diverse situazioni professionali e dati variabili di volta in volta?

Roberto Ferraris - Vercelli

Egregio signor Ferraris, ecco il nostro parere:

1) tutto il bene possibile; la FX-80 è una delle migliori stampanti della sua classe sul mercato, per giunta interfacciabile allo Spectrum con buona facilità (tanto per fare un esempio, il word processor "Tasword 2" è già predisposto per l'uso della Epson). Non ci risulta che esistano centri di assistenza tecnica specializzati, ma non dovrebbe essere difficile rintracciare un laboratorio che possa eseguire le eventuali riparazioni, data anche la notorietà e la diffusione del prodotto. Il suo innamoramento è pienamente giustificato, e poggia su ottime basi!

2) Se, come ci dice, vuole impiegare il calcolatore in particolare modo per il trattamento dei testi, la tastiera standard dello Spectrum risulta palesemente inadatta, tanto che vale sicuramente la pena di sostituirla. A seconda del modello, le varie tastiere esistenti sul mercato richiedono procedure di installazioni differenti, dal semplice incastro fino allo smontaggio completo dello Spectrum. Di regola non è necessario intervenire sulla scheda madre; nel peggiore dei casi chiunque abbia preso in mano una volta un saldatore non avrà difficoltà, e di norma la lista degli utensili necessari si ferma al cacciavite.

3) Dopo accese discussioni redazionali è stato stilato un piccolo elenco, ispirato da considerazioni di ordine qualitativo, che riportiamo in ordine rigorosamente alfabetico: Battle 1917 (della CCS), Galaxy Conflict (Martech), Hunter Killer (Protek), Special Operations (Lothlorien). A questi ci permettiamo di aggiungere i nostri "pallini", Lords of Midnight e Doomdark's Revenge, entrambi della Beyond, non esattamente definibili come simulazioni strategiche, ma ugualmente caratterizzati da una discreta complessità.

4) Se esiste, non è a livello professionale. Forse la cosa migliore da fare a questo riguardo è prendere un normale programma BASIC e riadattarlo in base alle proprie esigenze. Oltretutto, realizzare un programma di istogrammi non è difficilissimo, e tra i libri recensiti nelle pagine della rubrica PRINT sono già apparsi numerosi riferimenti all'argomento.

MISSIONE AFRICA

MIT

Spazio gratuito offerto da J.soft

**Una "Nave della Pace" in partenza dall'Italia
porterà in Africa soccorsi immediati e aiuti
per un domani migliore con il vostro contributo.**



La "Nave della Pace": uno strumento nuovo per intervenire immediatamente là dove è necessario un urgente soccorso a popolazioni gravemente colpite da calamità naturali. Non porta solo aiuti immediati ma anche strumenti di lavoro e mezzi ausiliari idonei a realizzare migliori condizioni di vita.

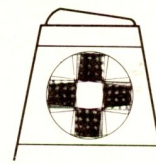
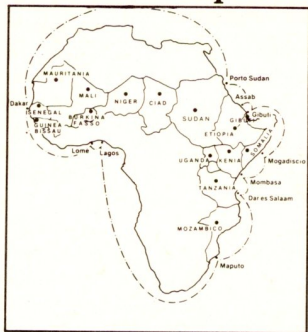
Affinché questo risultato sia raggiunto, gli aiuti siano adeguati ed efficaci e la "Nave della Pace" possa partire a pieno carico occorrono offerte e solidarietà da parte di tutti. Specialisti dei pro-

blemi dello sviluppo hanno offerto consigli per la migliore attuazione del progetto. L'iniziativa ha ottenuto l'Alto Patronato del Presidente della Repubblica Italiana, il

Patrocinio dell'ONU, della Lega di Società di Croce Rossa e di Mezza Luna Rossa e della Croce Rossa Italiana e il

contributo del Dipartimento per la Cooperazione allo Sviluppo del Ministero degli Affari Esteri. Il Comitato "Nave della Pace" chiede a quanti comprendono la necessità ed il valore dell'iniziativa, offerte di

mezzi e di beni indispensabili per i soccorsi di prima necessità e per gli aiuti di sviluppo destinati a 15 paesi dell'Africa.



Nave della Pace

COMITATO NAVE DELLA PACE - VIALE MAZZINI 41
00195 ROMA - TEL. 06/317447-386163

Da compilare in stampatello ed inviare in busta chiusa a:
Comitato Nave della Pace
V.le Mazzini 41 - 00195 Roma.

**SI', ANCH'IO VOGLIO AIUTARE
LA NAVE DELLA PACE A PARTIRE
CON LE STIVE PIENE**

Cognome _____

Nome _____

Via _____ N _____

CAP _____ Località _____

Per questo ho deciso di inviare il mio contributo di

- Lit. 10.000 Lit. 50.000
 Lit. 25.000 Lit. 100.000 o più

tramite:

Assegno non trasferibile intestato:
Comitato Nave della Pace

C/c postale n. 15285000

Bonifico bancario a credito
del c/c n. 3100/51 c/o
la Cassa di Risparmio di Roma
sede centrale -
Via del Corso, 320 - 00186 Roma

Desidero una ricevuta del mio versamento.





DI ELETTRONICA

DI UNA RIVISTA

MOLTO PIU'

**COMPUTERJYSTICK
ANALOGICO**

**UNITA PORTATILE
D'USCITA
30W E PIU'**

**Speciale
ANTIFURTO
DIGITALE**



Spedizione in Ab. Post. Gruppo III 79

GRUPPO
EDITORIALE
JACKSON

Nuova edizione della rivista
Mensile di elettronica pratica e hardware per microcomputer

ELETTRONICA 1

L.3000
MAGGIO 85

E' IN EDICOLA



QL, lo stato delle cose

di **Lucio Bragagnolo**

Sono passati ormai diversi mesi dalla prima comparsa del QL sul mercato inglese e italiano. Dopo molte vicissitudini pare si sia finalmente giunti a una versione definitiva o comunque destinata a non subire modifiche sostanziali a medio termine. Ecco un panorama riassuntivo della situazione.

JM, la definitiva

Come è abbastanza noto, per riconoscere la versione del proprio QL è sufficiente digitare

print ver\$

Dopo varie versioni, alcune note a tutti e altre quasi mitiche (come, in Italia, lo Spectrum Issue 1), è stata decisa quella definitiva, chiamata JM; delle altre ricordiamo i nomi AH, FB, PM, KB. Ha fatto una timidissima comparsa, a titolo sperimentale, anche una versione successiva alla JM, chiamata JS, installata su 89 calcolatori e poi subito abbandonata, in quanto peggiore delle precedenti. Le differenze tra le varie versioni sono piuttosto marginali, e consistono in una riduzione progressiva del numero di bug del sistema operativo QDOS, evidenti o nascosti. In più sono state adottate alcune modifiche hardware tese a migliorare le prestazioni dei microdrive e a ottenere un'immagine video soddisfacente anche usando un televisore. Si può dire che il funzionamento del QL JM sia perfetto: esistono ancora alcuni bug minori nel QDOS che però non pregiudicano assolutamente, in alcun modo l'attività "normale" di un possessore di QL (vale a dire l'uso in qualunque maniera dei microdrive, l'utilizzo dei quattro programmi applicativi forniti insieme al computer, la programmazione in SuperBASIC o in qualunque altro linguaggio alternativo di quelli disponibili, l'esecuzione - ov-

vamente! - dei programmi, il collegamento in rete locale, l'utilizzo delle varie porte di input/output presenti), esattamente come accade per lo Spectrum e per tutti gli altri computer esistenti. Dimenticavamo: chi volesse sostituire la sua ROM vecchia versione con la JM potrà spedire il proprio QL alla Sinclair per vederlo tornare aggiornato. È un peccato che il QDOS non permetta il multitasking da BASIC (comunque possibile in linguaggio macchina), anche se questa limitazione era stata ampiamente prevista dagli addetti ai lavori più competenti.

L'hardware interno

Il microprocessore è il Motorola 68008, fratellino minore della famiglia 68000, con architettura a 32 bit (anziché gli 8 dello Z80A Spectrum), coadiuvato nelle sue funzioni da un Intel 8049, che si occupa della tastiera, del suono e delle interfacce seriali RS232, lasciando al 68008 il compito di eseguire i programmi dell'utente e gestire il video. Sulla circuiteria non c'è molto da dire: sono state adottate soluzioni piuttosto economiche, nello stile Sinclair (basta pensare al prezzo del QL), che in ogni modo non vanno a scapito dell'affidabilità.

Abbiamo già provato varie volte a tenere acceso il QL per intere giornate, anche per constatare la precisione dell'orologio interno incorporato, che, non possedendo una propria batteria come invece accade nel Macintosh Apple, va inizializzato ad ogni accensione del computer. Il 68008 differisce dal 68000 nella struttura del bus dati esterno, che è di 8 bit anziché 16 o addirittura 32 come nel 68020, ultimo "parto" della Motorola per questa famiglia di microprocessori. In soldoni, la differenza si traduce in una riduzione di velocità di esecuzione, che rimane però ugualmente in misura incredibile sopra gli attuali standard degli home e personal computer almeno rimanendo

nel campo del linguaggio macchina (chi ha visto un Macintosh in funzione se ne immagini un equivalente un po' lento, in una misura che varia a seconda dell'uso più o meno intenso che viene fatto, appunto, del bus dati). Una ulteriore leggera riduzione è data dal fatto che il QL non possiede un coprocessore video, così che la gestione di quest'ultimo, come già detto, grava sulle spalle del 68000, che, ricordiamo, si trova a dover amministrare, nel caso peggiore, un totale di $512 \times 256 = 131072$ pixel rispetto ai $320 \times 200 = 64000$ del Commodore 64, ai $256 \times 192 = 49152$ dello Spectrum e dello standard MSX e i $512 \times 342 = 175104$ del Macintosh (che però ha un video monocromatico). Si può constatare l'effetto di questa soluzione usando una delle prime versioni del word processor Quill: i caratteri digitati (quindi "visti" dall'Intel 8049) appaiono sullo schermo (competenza, ricordiamo, del 68008) con un leggero ritardo, dando l'impressione che sia possibile perderne qualcuno per strada, cosa invece pressoché impossibile. Usando il linguaggio macchina, ripetiamo, si ottengono risultati impressionanti, mentre per il BASIC ovviamente le cose vanno in modo diverso (ne parleremo più avanti).

La RAM residente, composta da 16 chip installati nel QL, è di 128K, di cui 32 sono usati per lo schermo (a fronte dei quasi 7K dello Spectrum); eccettuata l'area delle variabili di sistema (imprecisata, ma comunque di estensione limitata) l'utente ha completamente a sua disposizione poco meno di 96K per i programmi (41K per lo Spectrum 48K, un massimo di 64K - che diventano 38K usando il BASIC - per il Commodore 64 e lo standard MSX, 128K - di cui 64 utilizzabili solo paginando la memoria - per Apple // e e // c, un massimo di 512 - teorici e mai effettivi - per il cosiddetto "Fat Mac", mentre per il Macintosh non espanso il discorso è identico, ma su una base di

128K). Attraverso espansioni esterne il 68008 può gestire direttamente, senza paginazioni o trucchi vari, la bellezza di un Megabyte (1024K per i profani). Solo placintosh offre prestazioni superiori – ben 16 Megabyte – mentre tutti gli home/personal computer con microprocessore a 8 bit non vanno oltre i 64K.

Il QL è ben fornito anche di porte di input/output: un connettore per espansioni di memoria, floppy o hard disk, un connettore per cartucce ROM, due connettori per joystick, due connettori RS232, la presa di collegamento col televisore, una presa RGB per monitor e due connettori per la rete locale (in pratica delle RS232 "povere"). Note dolenti? Mancano interfacce parallele, e le due porte seriali possono operare contemporaneamente, ma non con velocità di trasmissione/ricezione diverse (queste ultime sono comunque regolabili su diversi valori, da 75 a (solo in trasmissione 19200 baud). Per quanto riguarda il collegamento col monitor, a causa dell'alta frequenza di scansione occorre uno cosiddetto ad alta risoluzione, o sarà impossibile riuscire a visualizzare l'intero schermo generato dal QL senza che questo, letteralmente, esca dal video. Per finire ricordiamo la presenza di un tasto di reset distruttivo (equivalente al RUN USR 0 di Spectrum e ZX81) e, ahimè, l'assenza di un qualsivoglia interruttore.

L'hardware esterno

La tastiera del QL è stata fatta oggetto di molte critiche, secondo noi solo parzialmente giustificate. Sebbene infatti la digitazione non risulti agevole come ad esempio su tutte le keyboard Apple, il funzionamento è perfetto, e la velocità massima raggiungibile è comunque proibitiva per qualunque medio utilizzatore. Gli unici problemi potrebbero essere quelli di una velocissima dattilografa "a 10 dita" impegnata in operazioni di word processor. Le nostre riserve riguardano invece la posizione dei tasti CAPS LOCK e CTRL, che sarebbe stato molto meglio scambiare, visti gli equivoci che accadono ogni tanto digitando con una certa velocità e...distrazione. In più non viene segnalata in alcun modo la presenza o meno del CAPS LOCK, anche se le conseguenze di ciò sono più fastidiose che altro.

Altro soggetto di critiche ad oltranza sono i microdrive, alle quali anche qui ci allineiamo solo in parte. La nostra esperienza (solida e di lunga data) con i microdrive QL è stata tutto sommato positiva, confermando la nostra opinione secondo cui i microdrive, se usati con un minimo di attenzione, sono affidabili, tanto che da tempo abbiamo rinunciato al backup di ogni programma (eccettuati quelli forniti col QL), precauzione però che andrebbe presa almeno nel caso di file molto grandi o particolarmente importanti. L'unico in-

ABS	LIST
ACOS	LN
ACOT	LOAD
ADATE	LOCAl
ARC	LOG10
ARC__R	LRUN
ASIN	MERGE
AT	MISTake
ATAN	MOD
AUTO	MODE
BAUD	MOVE
BEEP	MRUN
BEEPING	NET
BLOCK	NEW
BORDER	ON... GOSUB
CALL	ON... GOTO
CHR\$	OPEN
CIRCLE	OPEN__IN
CIRCLE__R	OPEN__NEW
CLEAR	OVER
CLOSE	PAN
CLS	PAPER
CODE	PAUSE
CONTINUE	PEEK
COPY	PEEK__L
COPY__N	PEEK__W
COS	PENDOWN
COT	PENUP
CSIZE	PI
CURSOR	POINT
DATA... READ.. RESTORE	POINT__R
DATE	POKE
DATE\$	POKE__L
DAY\$	POKE__W
DEFine FuNction... END DEFine	PRINT
DEFine PROCedure.. END DEFine	RAD
DEG	RANDOMISE
DELETE	RECOL
DIM	REMark
DIMN	RENUM
DIR	REPeat... END REPeat
DIV	RESPr
DLINE	RETurn
EDIT	RETRY
ELLIPSE	RND
EOF	RUN
EXEC	SAVE
EXEC__W	SBYTES
EXIT	SCALE
EXP	SCROLL
FILL	SDATE
FILL\$	SDATE\$
FLASH	SElect... END SElect
FOR... NEXT... END FOR	SEXEC
FORMAT	SIN
GOSUB	SQRT
GOTO	STOP
IF... THEN... ELSE... END IF	STRIP
INK	TAN
INKEY\$	TO
INPUT	TURN
INSTR	TURNT0
INT	UNDER
LET	VER\$
LINE	WIDTH
LINE\$	WINDOW

fig. 1 - le parole chiave del SuperBASIC

conveniente accadutoci è stato un incomprensibile rifiuto del calcolatore di formattare – solo formattare – alcune

cartucce (ne abbiamo provate diverse) sul microdrive 2, rifiuto effettuato una ventina di volte, dopo di che il QL ha ab-

bandonato ogni remora, funzionando più che regolarmente. Le nostre uniche perdite di dati si sono verificate a causa di bug presenti nelle prime versioni dei quattro programmi (ne parleremo più avanti). È chiaro che il microdrive non è certo la soluzione migliore per un uso professionale o aziendale: si tratta comunque di due memorie di massa inserite nel computer che possono risultare utili per backup di emergenza o se, ad esempio, avete deciso di portarvi in vacanza il calcolatore e avete dei problemi di spazio: l'ingombro supplementare dei microdrive è uguale a zero, e le cartucce trovano comodamente posto nel più piccolo dei taschini (tra l'altro il loro costo, prima proibitivo, si è ridotto di recente - per chi legge, da qualche mese - a valori molto più adeguati). Ad ogni modo, saranno presto disponibili anche in Italia diversi modelli di disk drive disegnati per il QL, da 3"1/2 e da 5"1/4, in grado di accontentare anche i più esigenti. La velocità dei drive, per concludere, è leggermente superiore a quanto avviene nello Spectrum; la formattazione delle cartucce indica generalmente una capacità di 105/110K. C'è da dire che il sistema operativo restituisce il controllo all'utente molto prima che il drive abbia finito di lavorare, consentendo a volte grandi risparmi di tempo effettivo, e usa inoltre la RAM libera come memoria di supporto per il microdrive, così che la prima directory richiede qualche secondo per essere caricata, mentre quelle successive vengono caricate direttamente dalla RAM e sono quindi immediate. Lo stesso accade per i programmi: dando un NEW, questi non sono più rintracciabili in maniera normale, ma sono sempre presenti in memoria, tanto che, se richiamati, il drive gira solo per una frazione di secondo e il programma viene caricato istantaneamente, per l'appunto non dalla cartuccia ma dalla RAM.

II BASIC

Del BASIC, anzi SuperBASIC bisogna distinguere due aspetti: la sua concezione e l'effettiva resa alla prova dei fatti.

Per quanto concerne il primo punto non sapremmo dove appigliarci per fare una critica qualunque: il nostro giudizio è nettamente positivo. Il SuperBASIC è un linguaggio modernissimo,

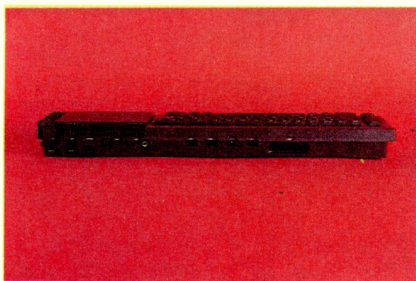


fig. 3 - visione posteriore del QL, con i vari connettori in evidenza

fig. 2 - i programmi disponibili per QL a tutto maggio 1985

nome	marca
QL Cash Trader	Accountancy Software of Torquay
Pascal Forth Assembler Typing Tutor Monitor	Computer One
Quilmerg Utility Pack	Pitch Associates
QL/APL	MicroAPL
Hungry Harry in the Haunted House	Snowsoft
QL Sprite Generator QL Super Backgammon QL Super Monitor	Digital Precision
QL Assembler 68K-OS C Compiler	GST Computer Systems
QL Touch'n'Go	Harcourt
Assembler BCPL Lisp	Metacomco
Hi-res Screen Dump Quill Abacus Easel Archive QL Chess	Positron Computing Psion
Business Accounts	Quest
QL Entrepreneur QL Project Planner QL Decision Maker QL Toolkit	Sinclair Research
QL Screen Editor Plus	S Gaymer
Solar Invaders Wall-Breakers Draughts Mined Your Path Statistical Averages Calendar	Equate
UCSD Pascal UCSD Fortran Advanced Development Toolkit	TDI Software
Sbutil Mbackup Terminal Chargen Sbextras FM FED DisASSEM NeatList	DataManagement Software

completamente strutturato, dotato di tutto il necessario e del più che superfluo per godere dei maggiori vantaggi del Pascal (grande leggibilità del programma, costruzione modulare) senza doverne sopportare la pesantezza sintattica e l'eccessivo, a volte, rigore formale. In più il SuperBASIC consente anche, a chi si accontentasse del BASIC normale, di ragionare regolarmente in termini di GOTO e GOSUB (inutili invece per chi sceglie la programmazione strutturata di DEFine PROCedure... END DEFine, IF... THEN... ELSE...END IF e SElect ON...END SElect), ed è per questo che sono rimasti i numeri di linea, altrimenti superflui. Esiste inoltre, all'interno del linguaggio, un mini Logo - con tanto di PENUP, PENDOWN, MOVE, TURN, TURNT0 -, un set completo di istruzioni grafiche di tutti i generi (manca solo una FILL diretta, compensata tuttavia da FILL 1 e FILL 0), la possibilità di spezzare gli intervalli dei cicli FOR...NEXT...END

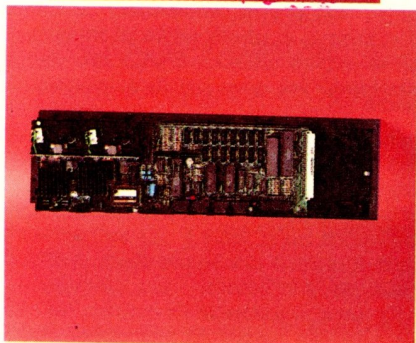
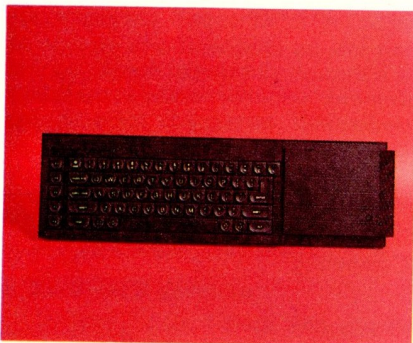


fig. 4,5,6 - la tastiera; un'immagine del manuale e del software fornito su microdrive; l'interno (notare in alto a sinistra i microdrive e a destra lo spazio per l'espansione di memoria).

QLDiary
QLLife
Databoss
Mailboss
Animal
Project
Screencopy
Dirmap
Freemem

Merry Muncher Blain Software
Fire Tower
Advance Invaders

Sort Softronic
Filedump

QL Forth R.E. Jackson
Shrub Bank and Planner SuperPlant Software
House Plant Bank and Planner

Space Trek Swansoft

Fantasia Adventure S.B. Software
Terminal Emulator QCode
68000 Assembler/Editor
3 Games

Executive Adventure Intersoft

Stockmarket Manager Portfolio Software

Agenda Q-Soft

Area Radar Controller Shadow Soft

Screen Dump Miracle Systems

QL Payroll TR Computer System
Full Screen Editor Bedsoft
Gambler
Beat the Clock
Auto-Draw
WD Utilities

WD Morse Tutor WDSsoftware
Ref QL

QL Art Eidersoft
QL Archiver
QL Dumps

Assembler Development
Package Adder

D-Day Games Workshop

The Lost Kingdom of ZQL Talent Computer Systems
WEST
GraphiQL

FOR o di RND (pensate alla comodità di avere un ciclo FOR i =1 TO 15 STEP 3, 17 TO 25, 44, 47, 55, 60 TO 100 STEP 2 o un RND (14 TO 299)!), tutto il necessario per gestire finestre, eseguire calcoli di ogni tipo (logaritmi naturali e in base 10, funzioni trigonometriche, divisioni, per modulo...), operare in memorie e sui file di program-

mi, di dati o di linguaggio macchina (basta accennare all'esistenza di tre tipi di POKE, da 8, 16 o 32 bit). Sono presenti anche istruzioni di autonumerazione e rinumerazione, di grande comodità e facilità d'uso: in definitiva siamo convinti che ben difficilmente chiunque riuscirà a padroneggiare il SuperBASIC tanto da riuscire a sfruttarlo nel modo

migliore e al 100%.

A proposito del secondo aspetto esiste invece qualche perplessità. Si sa che è tradizione Sinclair scrivere interpreti BASIC ottimi dal punto di vista concettuale, ma piuttosto lenti in esecuzione, e anche nel QL le cose vanno alla stessa maniera, anche se su un piano molto diverso di prestazioni da quello, poniamo, dello Spectrum. "Con un 68008 a 7.5 MHz di clock" - scriveva l'autorevole rivista americana "Byte" - "devo essere stati fatti degli sforzi per far andare il QL così piano". Il SuperBASIC è lento (intendiamoci bene: "lento" sta a significare, più o meno, una velocità media di elaborazione leggermente superiore a quella del Commodore 64) per alcuni aspetti e di una velocità quasi spaventosa in altri, rispecchiando peraltro alcune caratteristiche del 68008. Così l'elaborazione numerica è estremamente veloce: abbiamo provato a far eseguire al QL un benchmark suggerito sulla rivista "Creative Computing", dove venivano riportati i dati di 200 prove relative a circa 140 calcolatori di tutti i tipi. Il QL si è classificato al dodicesimo posto (PC IBM diciannovesimo, Olivetti M20 circa cinquantesimo, Atari 800XL centovesimo, Commodore 64 e Apple intorno alla centocinquantesima posizione, Philips MSX centosessantacinquesimo, Spectravideo e TI-99/4A centosettantesimi, ZX81 e Spectrum intorno al centoottantesimo posto). Altre attività, al contrario, vengono svolte con relativa lentezza, tanto che, esempio estremo, su un singolo ciclo FOR...NEXT di valore non elevato il Commodore 64 è più veloce. Altra osservazione negativa: alcune istruzioni del SuperBASIC sono soggette a limitazioni non citate nel manuale, come SELECT ON, che funziona solo su valori numerici e non su stringhe. Parlando infine della famosa "coercion" (la possibilità di scrivere $a\$=a+b\$$ quando $a\$$ e $b\$$ sono stringhe contenenti solo numeri, rendendo superflui gli operatori VAL e STR\$, inesistenti in SuperBASIC), non ci sentiamo di definirla negativamente o positivamente, in quanto non abbiamo ancora ancora svolto sul QL una mole di programmi tale da permetterci di capire se realmente sia un aiuto (come hanno detto alcuni nostri collaboratori) o una complicazione (secondo gli altri). Probabilmente l'opinione cambia a seconda del grado di esperienza e quindi di abitudine maturato su BASIC convenzionali.

Per riassumere, il SuperBASIC è un ottimo linguaggio per chi comincia o è digiuno di programmazione strutturata, ed è in grado di assolvere egregiamente le sue funzioni fino all'ambito piccolo-professionale. Per impieghi avanzati, particolari e ultraprofessionali è invece consigliabile l'uso di un altro linguaggio fra i tanti (praticamente tutti!) disponibili per il QL, vista l'assenza di grossi problemi dovuti alla eventuale carenza

di memoria - dato che anche solo 96K non sono poi così pochi -, e magari (perché no?) utilizzando il tutto in congiunzione con un sistema operativo alternativo disponibile su cartuccia ROM (ne sono già disponibili almeno due). Ah, ancora un paio di curiosità: nel SuperBASIC esistono due istruzioni supplementari. Una di queste è ELLIPSE, citata, ma non illustrata, nel manuale; l'altra appare solamente in un cantuccio del manuale ed è stata "scoperta" in contemporanea dai nostri collaboratori Stefano Pogliani ed Enrico Minetti: si tratta di MISTake, comando usato per far accettare all'interprete SuperBASIC linee di programma sintatticamente errate, nell'ambito di operazioni di debug, stesura, revisione dei programmi e di MERGE. Curiosando nella ROM è inoltre possibile rintracciare parole chiave probabilmente inserite nelle prime versioni e poi abbandonate, come ad esempio WHEN ERR. Questi "relietti" di epoca passata non sono tuttavia accettati dall'interprete. Chi trovasse altre keyword sconosciute ci scriva! La seconda curiosità è più nota, anche se sempre seminata sul manuale, ma non a tutti: in fase di EDIT, non date ENTER una volta finito di modificare una linea, ma "cursore giù" o "cursore su": vi verrà proposta per l'editing la linea successiva (o precedente), senza bisogno di dover digitare ogni volta EDIT...

Il manuale

Pur presentandosi bene esteticamente, ci sembra un po' sotto gli standard stabiliti con Spectrum e ZX81. A parte un foglio colmo di errata corrige, peraltro relativi solamente alla prima parte del manuale, l'impressione che si ricava è che si tratti di un manuale per principianti, da leggere come un libro di testo ma non da consultare. I singoli argomenti sono rintracciabili con difficoltà, e mancano sia una mappa della memoria dettagliata che un elenco delle principali variabili di sistema. A proposito, già che ci siamo ne diamo qualcosa, tanto per solleticare la curiosità dei già possessori di QL: potete vedere nel riquadro a pagina 16.

Abacus, Archive, Easel, Quill

Per quanto riguarda una recensione approfondita del package fornito dalla Sinclair insieme al QL vi rimandiamo a un prossimo numero, sia per motivi di spazio sia perché entreremo in possesso solo a giorni delle versioni 2.0 (cioè quelle definitive e di cui ci sembra più giusto parlare) di tutti e quattro i programmi. Per adesso riportiamo solo le prime impressioni; quelle positive riguardano la grande facilità d'uso data dalla ottima impostazione generale - uniforme per tutti i quattro programmi - e dalla vastità delle funzioni di Help, in pratica sostitutive del manuale. La velocità di elaborazione dati è mediamente buona; il programma che ci ha colpi-

to maggiormente è Easel, intuitivo nell'uso al punto di ricordarci il miglior software per Macintosh. Pure molto positiva è la facilità di installazione di caratteri di controllo per stampanti di ogni tipo (abbiamo provato finora con la Epson FX-80, la Diablo 630 con cui listiamo abitualmente i programmi e la Seikosha SP 800 I, dedicata espressamente al QL, già disponibile presso la Rebit e di cui riparleremo).

Le note negative sono frutto dell'uso dei programmi in vecchia versione (1.03 per Quill, 1.01 per gli altri) e con soli 128K di memoria, ricalcando in quest'ultimo caso gli stessi problemi che incontrano gli utenti di Macintosh non espanso a 512K. Brevemente, i difetti peggiori di queste versioni sono una spiccata tendenza a "piantarsi" quando uno meno se lo aspetta, il ritardo già menzionato tra l'input da tastiera e l'output su video, la lentezza di esecuzione di alcuni comandi, il continuo accesso ai microdrive (che determina noiose attese e perdite di tempo, oltre a favorire crash imprevedibili) e lo scarso ammontare di RAM libera, su cui però c'è da fare un discorso a parte. Il problema è identico a quello di Macintosh: il software che viene scritto è estremamente sofisticato, e in quanto tale richiede per forza grandi quantità di memoria, in misura tale da penalizzare l'utente che utilizza la versione base frequentissimi accessi a disco (Mac) o microdrive (QL), che diventano ovviamente sempre più rari man mano che viene aggiunta RAM supplementare. Un responsabile della Psion ha recentemente dichiarato che con un'espansione di soli 64K un utilizzatore di Quill versione 2.0 potrebbe scrivere oltre venti pagine di testo prima di dover "subire" un accesso del programma al microdrive (anche se il sistema operativo, come già detto, riesce molte volte a restituire il controllo della tastiera all'utente molto prima che il microdrive abbia smesso di girare). Con espansioni di 128, 256 o 512K un grafomane distratto potrebbe perfino riempire la memoria con più testo di quanto ne possa venire registrato su una cartuccia, e francamente non sappiamo se nei quattro programmi sia prevista o meno una tale eventualità, né tantomeno come essi si comporterebbero.

A parte questo, secondo la Sinclair le versioni 2.0 eliminerebbero tutti gli inconvenienti sopra descritti, oltre ad aumentare leggermente, grazie alla compattazione del programma, la memoria libera per i dati (attualmente, per fare un esempio, di 15K per Abacus e 8K per Easel). Ne riparleremo.

Espansioni hardware

Come è consuetudine dell'industria britannica, e come è già successo per lo Spectrum, già diverse aziende si sono fatte avanti con loro prodotti per chiudere i "buchi" hardware lasciati dai

progettisti Sinclair nel QL. Attualmente l'elenco del materiale disponibile prevede floppy disk da 3"1/2 e 5"1/4, hard disk da 5/10 Megabyte, interfacce parallele di tutti i tipi (di cui alcune consentono anche la selezione di valori diversi per le due porte seriali del QL, costruite una per le stampanti e l'altra per i modem), monitor ad alta risoluzione, espansioni di memoria, stampanti dedicate, modem, joystick, contenitori di cartucce microdrive (!), consolle porta-tutto e altro ancora. Di importante manca all'appello principalmente un connettore a più vie, dato che la maggioranza di questi prodotti è studiata per il collegamento tramite il connettore previsto dalla Sinclair per le espansioni di memoria.

Queste ultime, al momento, sono quelle che suscitano la maggiore perplessità, almeno per le più grandi, da 256K e 512K. Infatti, in attesa della "Wafer Scale Integration" promessa da sir Clive, in grado di condensare 512K in uno spazio ridottissimo, bisogna accontentarsi di grosse schede realizzate in maniera convenzionale, ingombranti e richiedenti alimentazione autonoma. Per il momento è probabilmente meglio orientarsi (anche per questioni economiche) su un'espansione di massimo 128K, peraltro più che sufficiente per la stragrande maggioranza delle applicazioni, anche considerando il fatto che questo limite corrisponde a poco più della massima quantità di dati memorizzabili in una cartuccia (supponiamo di avere un Quill che necessiti di 80/90K di memoria: dei 256K complessivi a disposizione ne rimangono circa 170, che diventano più o meno 140 considerando la memoria dedicata allo schermo. Una cartuccia può ospitare normalmente 105/110K). È chiaro che l'uso di un disk drive da 3"1/2 con una capacità di 400K sposta notevolmente i termini della questione...

Espansioni software

Dopo una certa qual sterilità iniziale, dovuta alle difficoltà ormai note, anche il mercato del software per QL comincia a muoversi, agevolato anche dal calo di prezzo delle cartucce microdrive, il cui alto costo scoraggiava gli investimenti delle software house. La direzione del movimento è però diversa da quella dello Spectrum, rivolgendosi in misura maggiore ai campi personal business, grafica, utilità e, soprattutto, linguaggi. Per il QL sono infatti già disponibili Assembler (dotato di un bellissimo full screen editor), Pascal, Pascal UCSD (uno degli standard più usati nel mondo), Forth, APL, BCPL, C, Lisp, Fortran e probabilmente ancora qualcun altro, che in questo momento ci sfugge o non è ancora uscito nel momento in cui scriviamo, come dire tutto quello che può servire a un utilizzatore evoluto (specialmente parlando di C e Pascal, i due linguaggi più utilizzati in questo periodo in campo professiona-

le). La grande quantità di memoria disponibile rende possibile anche la programmazione in Lisp, con tutte le implicazioni nel campo dei sistemi esperti e dell'intelligenza artificiale di cui questo linguaggio è sempre stato primo alfiere.

I programmi a carattere manageriale e finanziario sono tutti notevoli: citiamo qui Cash Trader, Project Planner, Decision Maker e Payroll come i migliori. Nel campo delle utility segnaliamo lo Sprite Generator, il sistema operativo CP/M 68K e l'Hi-res Screen Dump. Infine, tra i giochi facciamo i nomi di Super Backgammon, Radar Area Controller e lo stupendo QL Chess della Psion, vincitore dell'ultima campionato del mondo di scacchi elettronici e che non può che stupire chiunque con la sua visione tridimensionale della scacchiera. Bisogna dire, però, che, queste eccezioni a parte, l'ancora scarsa conoscenza del QL non ha finora portato a grandissime realizzazioni in campo software: il Manic Miner o il Knight Lore del QL sono di là da venire. E, pensando alle potenzialità del QL, immaginarci ciò che potrebbe essere realizzato ci fa correre un sottile brivido di piacere informatico lungo la schiena.

Un giudizio

Alla fine di questa (per forza di cose) sommaria, ma speriamo ugualmente interessante carrellata sul QL, il nostro giudizio è sostanzialmente positivo, anche se permangono su di esso alcune macchie d'ombra. Il QL si propone come una specie di Spectrum super evoluto: pensandoci, sono stati potenziati o risolti tutti quegli aspetti sui quali lo Spectrum presentava carenze o inadeguatezze dovute all'...età (tastiera, BASIC strutturato e veloce, memoria, risoluzione grafica, memoria di massa, li-

mitazione di colore - nel QL c'è completa libertà di colorazione dei singoli pixel), editor di linea, assenza di routine di renumber o autonumber e via dicendo. D'altro canto, la ricerca di soluzioni economiche ha portato a penalizzare questa macchina rispetto ai possibili contendenti nel campo del personal business: un computer senza un interruttore non fa certo una buona impressione. La nostra opinione è che, usato con un disk drive, espansione di memoria - facoltativa -, una buona stampante, un monitor ad alta risoluzione e l'uso di un linguaggio alternativo (e, si badi bene, tutto questo è già disponibile in Italia: dovrebbe essere in vendita tra breve presso la Rebit un disk drive da 3"1/2 perfettamente compatibile col QDOS e dotato del software necessario a copiare su disco i quattro programmi Psion, successivamente richiamabili dall'interno di uno stesso menu), congiunto al SuperBASIC e a un Assembler, il QL possa validamente proporsi come un sistema dal rapporto prezzo/prestazioni eccezionalmente competitivo rispetto a qualsiasi altro concorrente. Il grande punto di forza di questo calcolatore è il suo microprocessore, e questo fa ben sperare per il futuro in termini di software, una volta superata l'inevitabile fase di stallo che accompagna la vendita di un numero di computer sufficiente a destare l'interesse delle software house, fase che anche lo Spectrum e il Commodore 64 (e perfino l'Apple II) hanno dovuto attraversare, per citare le tre macchine col più ampio parco programmi esistente al mondo. E in ogni caso, se non si volessero affrontare spese eccessive, per un milione si può portare a casa uno dei più potenti home computer sul mercato, completo di una memoria di massa più che sufficiente per l'uso diciamo casalingo, in

POKE 163891,1

"congela" lo schermo. Lo stesso risultato si ottiene schiacciando contemporaneamente CTRL e F5, combinazione utilissima per interrompere momentaneamente un LIST, fermare il movimento di una immagine per scattare una foto, andare un momento in bagno...; una successiva pressione di F5 o di un altro tasto riattiva lo schermo, come accade dando POKE 163891,0

PRINT PEEK (163890)

vi dice se state lavorando in modo monitor (0) o TV

POKE W 163976,255

POKE W 163976,0

POKE W 163980,n

attiva CAPS LOCK
disattiva CAPS LOCK

altera il ritardo con cui un tasto premuto entra in autorepeat (n è espresso in decimi di secondo)

POKE W 163982,n

altera la velocità di autorepeat. Se vi capita di assegnare 0 a entrambi gli ultimi due valori, vi sconsigliamo di tentare degli input da tastiera...

PRINT PEEK (164078)

vi dice quale microdrive è sotto il controllo del sistema operativo

PRINT PEEK (164895)

se siete collegati in rete, vi dà il numero di stazione del vostro QL.



grado di funzionare, se necessario, anche su un comune televisore, completo di un package di programmi a livello professionale. Oltretutto è molto facile che il prezzo del QL sia destinato a scendere ulteriormente, soprattutto se avverrà il previsto lancio in Italia di un QL finalmente ufficializzato, sostenuto dalla Sinclair Italia e in versione assolutamente definitiva: è di questo periodo la notizia dell'arrivo della nuova serie di computer Atari, completi di mouse, tastierino numerico, disk drive da 3"1/

2, gestione video a finestre e software stile Macintosh al prezzo di 700 sterline (contro le 399 del QL).

Tutto sommato, il QL ha le carte in regola per conquistare sul mercato italiano un buon successo: ma è ancora molto presto per fare previsioni sicure. Da parte nostra pensiamo di aver dato attraverso queste notizie, un contributo utile per una valutazione personale di quale sia l'effettiva validità del QL e di che livello siano le sue prestazioni.

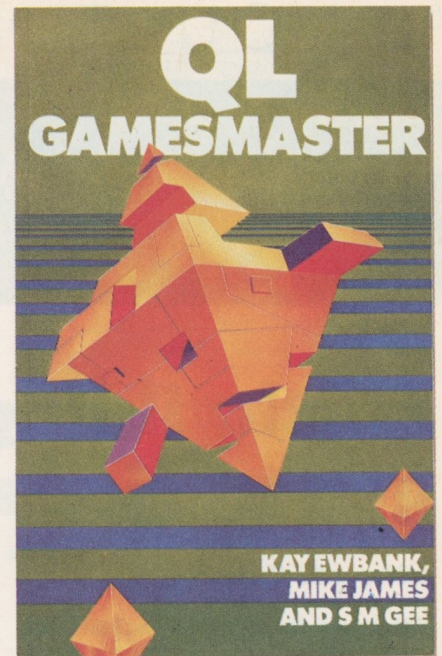


fig. 7 - Uno dei numerosi libri sul QL disponibili.

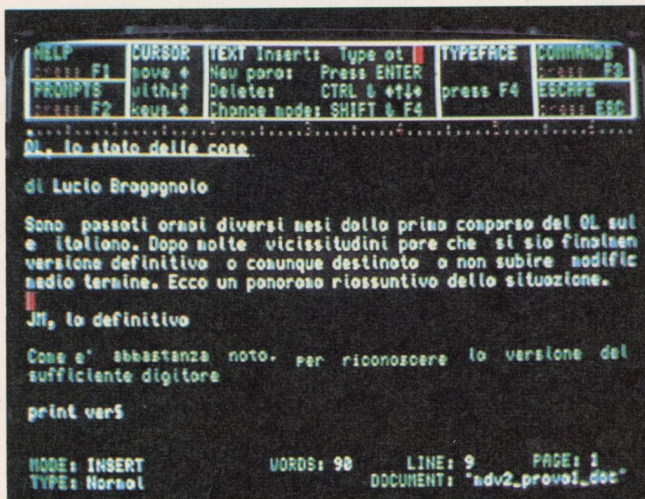


fig. 8 - L'inizio di questo articolo, realizzato con QUILL. Notare il grassetto e la sottolineatura, visibili anche sullo schermo.

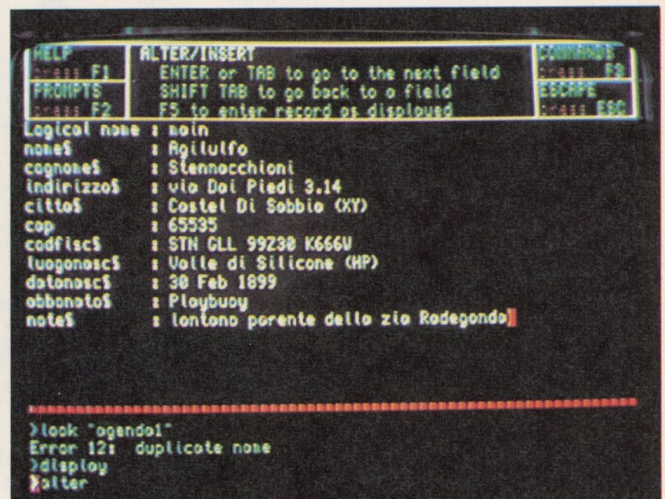


fig. 9 - ARCHIVER è un data base flessibile e di discreta potenza.

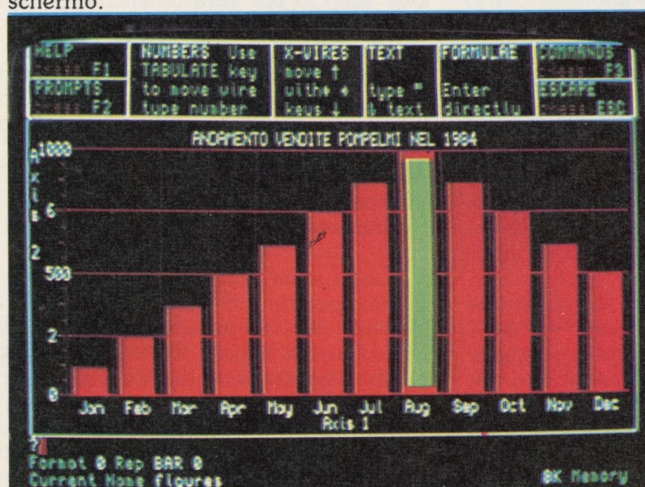


fig. 10 - Un semplice grafico realizzato con EASEL, forse il migliore dei quattro programmi per immediatezza e facilità d'uso.

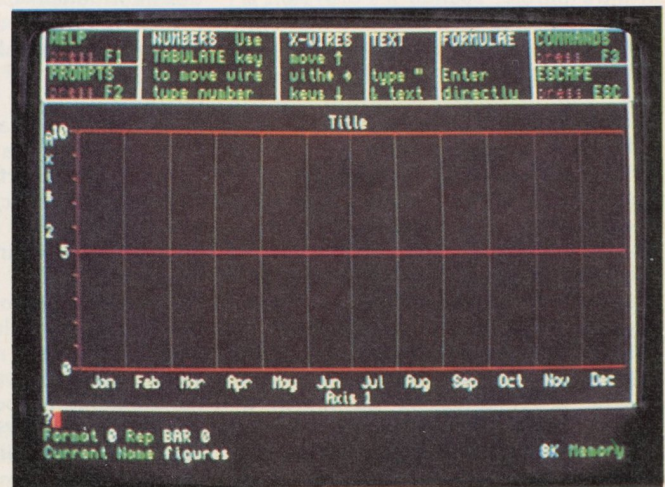


fig. 11 - I comandi rimangono sempre gli stessi da programma a programma. Qui vediamo una tipica schermata dello spreadsheet ABACUS.

...then

Forth: il linguaggio della quarta generazione

di **Marcello Spero**

Nato per impieghi specialistici, è oggi diffusissimo anche a livello di home e personal computer...compreso lo Spectrum!

Come secondo linguaggio della nostra serie abbiamo scelto il Forth. Non si tratta di una scelta casuale. Questo linguaggio, nato per scopi piuttosto specialistici, e destinato perciò a una ristretta fascia di utenza, nel corso dei suoi quindici anni di vita ha riscosso un interesse enorme, che lo ha portato ad una diffusione praticamente universale nel mondo della microelaborazione.

Questa sua diffusione è dovuta, oltre che alle sue peculiari caratteristiche di sintassi, ad una singolare "predisposizione" all'uso su piccoli e piccolissimi sistemi, adattandosi sorprendentemente alla loro struttura hardware.

Riprendendo il discorso fatto qualche tempo fa in queste pagine, possiamo dire che fra tutte le "macchine virtuali" create dai vari linguaggi, quella del Forth è la più simile alla macchina reale, cioè all'hardware del computer.

In altre parole, usando il Forth non si perde di vista la vera struttura del computer che lo ospita.

Ma vediamo le cose dall'inizio.

In principio

Il Forth è stato inventato attorno al 1970 da Charles H. Moore, allora al National Radio Astronomy Observatory. L'obiettivo era di creare un linguaggio per il controllo di processo, in particolare per il controllo dei telescopi.

Considerato dal suo creatore un linguaggio della quarta generazione, il Forth prende il suo nome dall'aggettivo inglese "fourth", che significa appunto quarto. L'abbreviazione da fourth a forth è dovuta al fatto che Moore sperimentava il suo nuovo linguaggio su uno dei primi calcolatori interattivi, l'IBM 1130, in grado di accettare solo cinque caratteri identificatori.

All'inizio il Forth è stato impiegato so-

prattutto per simulazioni in tempo reale, cui è adatto per la sua grande velocità. I miglioramenti introdotti nel corso degli anni successivi ne hanno permesso l'uso e la diffusione anche in altri campi dell'informatica. Come abbiamo detto, è stato adottato per molte applicazioni su mini e microcalcolatori, anche perché i programmi Forth occupano poca memoria.

A differenza di quanto accade con altri linguaggi, primi fra tutti Logo e Pascal, i programmi in Forth talvolta sono, data la loro grande concisione, di difficile lettura. Molte parole chiave di questo linguaggio sono addirittura segni di interpunzione.

Nel Forth il dispositivo fondamentale del calcolatore è la "catasta" o "pila" (stack), un'area di memoria organizzata come una pila di vassoi, cosicché il primo elemento introdotto sarà l'ultimo a poterne essere estratto. Poche basilari nozioni di Assembler sono sufficienti a comprendere come questa struttura "virtuale" si adatti a meraviglia ad una struttura realmente esistente in tutti i microcomputer, lo stack appunto.

I calcoli "a rovescio"

Pur apparentemente contorto nella forma, il Forth rivela immediatamente la sua struttura estremamente semplice e chiara.

Due sono le sue caratteristiche fondamentali, che lo rendono diverso da tutti gli altri linguaggi: il Forth è basato sul concetto di "parola", ed utilizza la "notazione polacca inversa", che opera sulla catasta. Questo a differenza di linguaggi come il BASIC, che è basato sul concetto di linea e numero di linea ed utilizza la notazione algebrica per l'introduzione di istruzioni di tipo aritmetico.

Vediamo innanzitutto cosa intendiamo quando parliamo di "notazione polacca inversa" (molto conosciuta con la sua abbreviazione inglese RPN). In BASIC per far eseguire la somma di due numeri, supponiamo 5 e 7, e poterne conoscere il risultato dobbiamo scrivere

```
PRINT 5+7
```

Questo modo di procedere corrisponde a quello cui siamo stati abituati a scuola,



vale a dire la notazione algebrica. In Forth, viceversa (è proprio il caso di dirlo), la stessa operazione va espressa così:

5 7 + . ;

a significare

5 prendi il 5
7 prendi il 7
+ sommi
. stampa la somma
; fine programma

Raffigurandoci la catasta come una struttura orizzontale, possiamo raffigurarci il meccanismo di azione dell'operazione in questo modo:
situazione iniziale

5

5

7

75

+

12

Possiamo ora riscrivere la spiegazione dell'operazione appena vista, alla luce di quanto abbiamo capito:

5 metti sulla catasta il numero 5
7 metti sulla catasta il numero 7
+ somma i primi due numeri sulla catasta, eliminali dalla catasta, metti sulla catasta il risultato
. stampa il risultato

Il segno "+" ed il segno "." sono a tutti gli effetti "parole" del Forth. Stanno ad indicare cioè una determinata sequenza di operazioni. In Forth, in effetti, esistono solo due tipi di "oggetti": le parole e i numeri. Le prime indicano una sequenza di operazioni, i secondi vengono posti sulla catasta. A riprova del fatto che in una catasta "gli ultimi saranno i primi", proviamo

questa sequenza:
3 4 5 6 . ;
Il numero stampato sarà 6. Infatti:

3

3

4

4 3

5

5 4 3

6

6 5 4 3

5 4 3

Se ora dessimo
. ;
otterremmo sullo schermo prima 5, poi 4 ed infine 3.
Proviamo con qualcosa di più complesso:

4-2*7+5(2+6)

Questa espressione andrà introdotta con i numeri nell'ordine inverso a quello in cui dovranno essere utilizzati, mentre i segni, o meglio le parole, nel giusto ordine sequenziale. Questo considerando che per la sua stessa natura la notazione polacca inversa non dispone di parentesi. Quindi:

ordine di esecuzione

1	2+6
2	5*
3	7+
4	2*
5	4-

e perciò:

4 2 7 5 2 6 + * + * - ;

Seguiamo il procedimento nel solito modo: eccola; situazione dopo l'introduzione di tutti i numeri:

6 2 5 7 2 4

+

8 5 7 2 4

*

40 7 2 4

+

47 2 4

*

94 4

-

98

A questo punto, volendo conoscere il risultato, potremo dare

. ;

in alternativa potremo proseguire con altre operazioni, che utilizzino il risultato appena ottenuto.

E se volessimo ambedue le cose? In BASIC la cosa si risolverebbe con

LET a=4-2*7+5*(2+6)
PRINT a
LET a=....

utilizzando cioè una variabile per conservare il valore del risultato. In Forth possiamo fare così:

4 2 7 5 2 6 + * + * - DUP . ;

Il calcolo avviene allo stesso modo di prima, ma una volta giunti al risultato la parola DUP ha questo effetto:

94

DUP

94 94

...then

DUPlica cioè il primo elemento della catasta. In questo modo possiamo utilizzare uno dei due elementi per la stampa, e l'altro resterà memorizzato per calcoli futuri.

Già questo semplice esempio dà un'idea di come questo tipo di matematica renda secondarie le variabili.

Non che in Forth queste non esistano: semplicemente, il loro uso è piuttosto occasionale.

Di parole che eseguono spostamenti o copie degli elementi in catasta ne esistono parecchie:

DUP duplica il primo elemento
 DROP elimina il primo elemento
 SWAP scambia fra loro di posto i primi due elementi
 ROT ruota la posizione dei primi tre elementi

e così via.

Ora che ci siamo fatti un'idea sul funzionamento della matematica RPN, torniamo al concetto di parola.

Tutti i comandi e le istruzioni Forth, abbiamo detto, sono parole. Ci sono le parole matematiche, che comprendono i segni delle operazioni, le funzioni di uso corrente quali SQR, MOD, ecc., quelle logiche, quelle per la manipolazione della catasta, quelle per la formattazione della stampa, e così via.

Uno dei grandi vantaggi del Forth è comunque quello di consentire all'utilizzatore di ampliare a piacere questo vocabolario, con parole create da lui per mezzo di associazioni di parole già esistenti.

Se ad esempio definiamo la parola

DOPPIO

per ottenere il doppio di un numero presente al primo posto sullo stack come

2 * ;

potremo in un secondo tempo definire la parola

QUADRUPLO

come

DOPPIO 2 * ;

e la parola SESTUPLO

come

DUP QUADRUPLO SWAP DOPPIO + ;

Perplessi?

Proviamo a seguire il meccanismo di SESTUPLO, supponendo di volerlo calcolare per il numero 3. Avremo:

3

DUP

3 3

QUADRUPLO

DOPPIO

6 3

2

2 6 3

*

12 3

SWAP

3 12

DOPPIO

6 12

+

18

In questo modo la programmazione strutturata diviene una realtà relativamente semplice e dagli evidenti vantaggi.

C'è poi da dire che queste nuove parole sono riutilizzabili senza alcun problema in qualsiasi altro programma. Non esiste, infatti, problema di corrispondenza di variabili, visto che viene utilizzata la catasta per il trasferimento dei dati. Basterà che questi siano nel giusto ordine. Proprio per sfruttare a fondo l'estrema flessibilità delle nuove parole create dall'utilizzatore, praticamente tutti i

Forth prevedono la possibilità di memorizzarle su cassetta, disco o microdrive, a seconda della macchina.

In questo modo, col tempo si creerà una vera e propria libreria di parole (vere e proprie routine, in realtà) per far fronte a tutte le necessità.

Ripetiamo ...

Abbiamo parlato di notazione matematica, di parole e di programmazione strutturata. Un linguaggio non può però dirsi tale senza appropriate strutture di ripetizione e di controllo, che consentano la creazione di cicli con controllo a contatore o a condizione. Sempre sulla base di determinate condizioni deve essere possibile effettuare delle scelte.

In questo campo le strutture del Forth ricordano molto da vicino quelle del Pascal, ferma restando la sua sintassi "a testa in giù".

Troviamo infatti la struttura

DO...LOOP

per realizzare cicli a contatore. Il suo uso è molto semplice:

- al primo posto in catasta dovrà esserci il valore di partenza del contatore
- al secondo posto il valore finale.

Ad esempio:

10 1 DO "pippo" LOOP ;

stamperà per dieci volte la stringa "pippo". Tra parentesi, come vedete è possibile la stampa di una stringa delimitata da apici, che va però posposta alla parola di stampa.

E se volessimo utilizzare, all'interno di un ciclo, il valore del contatore?

Anche questo è possibile. Il contatore di un ciclo DO...LOOP è sempre e solo la variabile I, o meglio: la parola I produce come risultato al primo posto in catasta il valore corrente del contatore. Quindi:

10 1 DO I . LOOP ;

stamperà i numeri da 1 a 10.

La struttura

BEGIN...UNTIL

serve invece a realizzare cicli con controllo a condizione.

BEGIN 2 + DUP 8 = UNTIL ;

...then

Forth: il linguaggio della quarta generazione

continua a sommare 2 al numero inizialmente al primo posto in catasta, finché il risultato non sia uguale a 8. La duplicazione del risultato prima di sottoporlo alla condizione è necessaria, in quanto quest'ultima per disporre degli oggetti del confronto li toglie dalla catasta. Esiste anche un ciclo

BEGIN...AGAIN

senza possibilità di uscita, e un

BEGIN...WHILE...REPEAT

che effettua un test in corrispondenza della parola WHILE; se il risultato è 1, cioè vero, prosegue fino a REPEAT e quindi riprende da BEGIN, altrimenti il ciclo viene troncato e si passa all'eventuale istruzione successiva a REPEAT. Come vedete, la disponibilità di strutture è in questo campo addirittura ridondante.

Per quanto riguarda le condizioni vere e proprie, troviamo la classica

IF...THEN

e la più complessa

IF...ELSE...THEN

Il funzionamento segue sempre lo stesso principio "al contrario":

9 > IF DROP THEN ...

se non risulta maggiore del numero precedentemente al primo posto in catasta, verrà eseguita la parola DROP, e quindi si proseguirà. In caso contrario l'esecuzione salterà oltre la parola THEN.

9 > IF DROP ELSE 2 + THEN...

Alla situazione precedente si è aggiunta qui una nuova possibilità: se la condizione

9 > numero

non è verificata, l'esecuzione passerà alle istruzioni contenute fra le parole ELSE e THEN.

È superfluo precisare che fra IF ed ELSE, fra IF e THEN e fra ELSE e THEN può essere inserito un numero qualsiasi di istruzioni.

Per concludere questa necessariamente incompleta esposizione, vediamo un programmino che esegua un compito semplice, scritto prima in BASIC e poi

in Forth.

Si tratta di una routine che somma fra loro solo gli elementi di un vettore aventi valore dispari.

Ecco il listato in BASIC:

```
10 DIM t(50)
20 READ n
30 FOR i=1 TO n
40 READ t(i)
50 NEXT i
60 GO SUB 1000
70 DATA 4
80 DATA 23,34,7,9
1000 LET s=0
1010 FOR i= 1 TO n
1020 IF INT(t(i)/2)*2 <> t(i) THEN
      GO TO 1040
1030 LET s=s+t(i)
1040 NEXT i
1050 RETURN
```

e in Fort:

```
: SOMMADISPARI
0 SWAP 0
DO
  SWAP DUP 2 MOD
  IF +
  ELSE DROP
  THEN
LOOP
```

e, per far partire il programma con i dati desiderati:

23 34 7 9 4 SOMMADISPARI . ;

Prima di tutto c'è da fare una precisazione. La stesura "indentata" del programma (cioè rientrante e verticale) non è possibile con tutti i Forth in commercio. Noi l'abbiamo adottata qui perché rappresenta il modo più chiaro di scrivere un programma del genere.

E veniamo al confronto fra i due programmi.

In BASIC è stato necessario dimensionare in eccesso un vettore, supponendo di non conoscere in anticipo il numero dei dati.

Il programma, dopo una sequenza di assegnazione dei valori, chiama una subroutine che si occupa di sommare fra loro i valori dispari. Quindi si ritorna al programma principale, che effettua la stampa del risultato.

Le variabili utilizzate sono:

t per il vettore
n per il numero dei dati
s per la somma
i come contatore

Il programma Forth inizia con il simbolo ":", che significa: "la parola che segue è il nome del blocco che la segue, fino al punto e virgola. L'esecuzione non sarà immediata, ma il blocco verrà memorizzato".

Per comprenderne il funzionamento occorre seguirne la meccanica in presenza dei dati, quindi dopo la chiamata da parte di

23 34 7 9 4 SOMMADISPARI . ;

la situazione sarà:

4 9 7 34 23

0

0 4 9 7 34 23

SWAP

4 0 9 7 34 23

0

0 4 0 9 7 34 23

DO (utilizza 0 e 4 come estremi)

0 9 7 34 23

SWAP

9 0 7 34 23

DUP

9 9 0 7 34 23

2

2 9 9 0 7 3 4 2 3

MOD (calcola il resto di 9/2)

1 9 0 7 3 4 2 3

IF (trova 1 in cima: verificato)

9 0 7 3 4 2 3

+

9 7 3 4 2 3

LOOP

DO

16 34 23

SWAP

34 16 23

DUP

34 34 16 23

2

2 34 34 16 23

MOD

0 34 16 23

IF (il primo è 0: non verificato)

34 16 23

ELSE

DROP

LA CULTURA INFORMATICA

...then



16 23

THEN

LOOP

DO

16 23

SWAP

23 16

DUP

23 23 16

2

2 23 23 16

MOD

1 23 16

IF

23 16

+

39

THEN

LOOP

;

Non sono state utilizzate variabili, e l'oc-

cupazione di memoria, in questo caso di catasta, è stata limitata alle reali necessità.

Dunque?

È sempre difficile dare un giudizio su di un linguaggio. Per questo ci limiteremo a sintetizzarne pregi e difetti principali.

Il Forth è semplice da imparare e da usare, flessibile ed estremamente veloce.

Contiene inoltre l'occupazione di memoria a livelli davvero minimi.

D'altra parte i programmi lunghi diventano intricati e poco comprensibili. Per questo è sempre consigliabile fare un intenso uso delle possibilità di strutturazione, che si traducono in programmi formati da parole formate da altre parole e così via.

Altra limitazione importante riguarda l'aritmetica, solo intera, che rende questo linguaggio non adatto ad applicazioni strettamente matematiche.

Per quanto riguarda la disponibilità commerciale di Forth per lo Spectrum, la scelta è piuttosto ampia. Oltre al Forth compilato, della Sigma Technical Press, notevole e veramente veloce (probabilmente il più veloce linguaggio disponibile per Spectrum, anche se per nulla standard), sono molti i prodotti interessanti. Noi abbiamo provato l'interprete (è questa la forma normale per il Forth) della CP Software, completo e totalmente standard, a parte alcune estensioni per rendere disponibili con facilità dall'interno del linguaggio le possibilità grafiche e sonore dello Spectrum.

REM:HW

Hardware

Miscellanea

di **Marcello Spero**

Alcuni "piccoli grandi suggerimenti", tra i più richiesti per Spectrum e ZX81

Grandi novità, questa volta! Sospendendo, per il momento, il capitolo dedicato all'interfaccia (non vi preoccupate, prestissimo troverete in queste pagine ulteriori aggiunte, applicazioni e modifiche alla nostra ormai storica interfaccia), passiamo ad un nuovo argomento.

In questa, e nelle puntate che seguiranno, ci occuperemo di migliorare in tutti i modi le prestazioni delle macchine Sinclair (certo: non si parlerà soltanto di Spectrum). In qualche caso si tratterà di rimediare a difetti, noti e meno noti. L'adeguamento dei "vecchi" Spectrum alle prestazioni del Plus sarà un altro obiettivo. È infine in preparazione una serie che riguarda l'uso dello Spectrum per scopi professionali, e di conseguenza i problemi e le possibili soluzioni (tastiera professionale, collegamento con stampanti non ZX, ecc.).

Speriamo inoltre di poter inserire in questa sede le risposte ai molti quesiti riguardanti vari argomenti di hardware che ci giungono da ogni parte.

Plus o non Plus?

Ed è proprio con la risposta ad un interrogativo che apriamo la prima puntata di questa nuova serie.

L'apparizione dello Spectrum Plus ha suscitato generosi (e meritati) commenti di approvazione, ma anche qualche dubbio riguardante la sua effettiva compatibilità con il "vecchio" fratello, nonché sul come utilizzare i nuovi tasti all'interno dei propri programmi.

Per quanto riguarda il primo problema, la prova dei fatti ha dissipato ogni perplessità. È addirittura possibile, per gli utenti Spectrum di vecchia data duri a convertirsi, l'uso dei vecchi tasti shiftati in luogo dei nuovi (ad esempio, M e SYMBOL SHIFT al posto del punto), anche se sui primi non c'è più l'indicazione di quei simboli ora assurdi alla glo-

ria di un tasto indipendente. L'uso dei nuovi tasti, ed in particolare dei cursori, in programmi realizzati da noi, come pure la modifica dei programmi commerciali scritti in BASIC, richiede un discorso a parte. A proposito: i programmi commerciali scritti in BASIC sono pochi, purtroppo; certo, in teoria è possibile la modifica anche di quelli in linguaggio macchina, ma il tempo necessario per individuare, all'interno della "selva" di codici, la routine di lettura della tastiera è tale da scoraggiare anche i più testardi.

Dunque, dicevamo che il vecchio sistema di utilizzo dei tasti cursore, basato su linee tipo

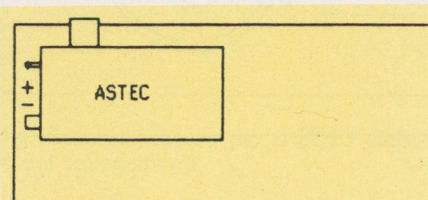


fig. 1: collegamenti all'interno dello Spectrum per ottenere il segnale video per monitor

```
10 IF INKEY$="5" THEN...
20 IF INKEY$="6" THEN...
30 IF INKEY$="7" THEN...
40 IF INKEY$="8" THEN...
```

rende utilizzabili gli ex-cursori (quelli appunto sui tasti 5,6,7,8), ma non i nuovi tasti.

I nuovi tasti sono infatti "autoshiftanti"; dispongono cioè di un doppio contatto, e si comportano perciò come due tasti in uno: lo shift ed il relativo cursore. Per programmarne correttamente l'uso, occorre perciò tener presente che:

cursore a sinistra = CHR\$ 8
 cursore a destra = CHR\$ 9
 cursore in basso = CHR\$ 10
 cursore in alto = CHR\$ 11

e comportarsi di conseguenza. Le linee necessarie saranno quindi

```
10 IF INKEY$=CHR$ 8 THEN...
20 IF INKEY$=CHR$ 9 THEN...
30 IF INKEY$=CHR$ 10 THEN...
40 IF INKEY$=CHR$ 11 THEN...
```

Lo stesso tipo di modifica può essere eseguita all'interno di programmi acquistati o copiati da riviste, che contengano riferimenti ai "vecchi" tasti cursore. Questo non tanto per snobismo nei confronti dello Spectrum "non Plus", ma perché la posizione dei nuovi cursori è decisamente più comoda.

Monitor!

La seconda proposta riguarda invece tutti i possessori di Spectrum, Plus o non Plus. Si tratta infatti di un sistema per utilizzare come unità video un monitor od un televisore, provvisto di presa monitor (viene chiamata normalmente SCART).

Si tratta, molto semplicemente, di collegare un tratto sufficientemente lungo di cavo schermato all'ingresso del modulatore video. Procedete così:

- aprite il vostro Spectrum, togliendo le viti che trovate sul fondo;
- facendo attenzione a non distaccare i contatti che collegano la tastiera al resto del sistema, sollevate e fate scivolare indietro la parte superiore della macchina quanto più è possibile;
- guardando il circuito, noterete in alto a destra una scatoletta metallica con la scritta ASTEC: si tratta del modulatore. Vi saranno chiaramente visibili, sul suo lato sinistro, due fili che entrano nel contenitore metallico;
- di questi due fili quello verso di voi è il polo negativo, mentre quello in alto (o più indietro, se preferite) è il positivo;
- prendete il cavo schermato e collegatene la calza (cioè il conduttore intrecciato a maglia di forma tubolare che sta subito sotto il rivestimento esterno) al

negativo, cioè quello verso di voi;

- il conduttore interno, sottile, va collegato invece al positivo;
- l'unico modo di eseguire questi collegamenti è la saldatura. Per non rovinare il modulatore questa dovrà essere veloce; se non vi sentite in grado, affidate questo compito a una persona esperta;
- all'altra estremità del cavo andrà collegata una spina, diversa a seconda del tipo di apparecchio cui ci si deve connettere;
- i monitor hanno normalmente un ingresso uguale a quello di un televisore, ossia provvisto di presa ad imbocco coassiale. Il polo positivo andrà collegato al contatto centrale, quello negativo al contatto periferico;
- la presa SCART dei televisori a colori di recente fabbricazione richiede invece un connettore di tipo EURO a 20 contatti. In questo caso, dopo aver acquistato da un rivenditore di materiale televisivo una spina adatta, bisognerà collegare il contatto 20 al polo positivo del nostro cavo e il 18 o 17 al negativo;
- in entrambi i casi, se l'immagine non appare, il primo tentativo da fare, naturalmente dopo aver controllato saldature e collegamenti, è l'inversione dei poli.

Una precisazione, rivolta in particolare a chi non è proprio digiuno di queste cose. Quello che preleviamo all'ingresso del modulatore è un segnale composto PAL. I monitor adatti sono quindi quelli PAL, e non gli RGB. Il collegamento con questi ultimi, che darebbe risultati ancora migliori, è reso complesso dal fatto che al circuito video dello Spectrum non giungono, come di consueto, i tre segnali

R = rosso
G = verde
B = blu

ma due segnali di differenza, e precisamente

U = blu-giallo
V = rosso-giallo

Questa situazione richiederebbe, per ottenere i tre segnali RGB, una complessa decodifica. Chissà, forse un giorno ce ne occuperemo...

Utile per tutti è invece sapere che il tipo di collegamento da noi suggerito va bene sia nel caso di monitor a colori che in bianco e nero.

Attenzione, ottantunisti!

Collegarsi al modulatore può, stranamente, essere di aiuto anche nel miglioramento della registrazione dei programmi su cassetta.

Questa modifica riguarda lo ZX81, macchina che presenta purtroppo qualche problema in questo campo. Il suo effetto è la creazione di un segnale acuto, che precede l'inizio della trasmissione dei dati, come avviene per lo Spectrum.

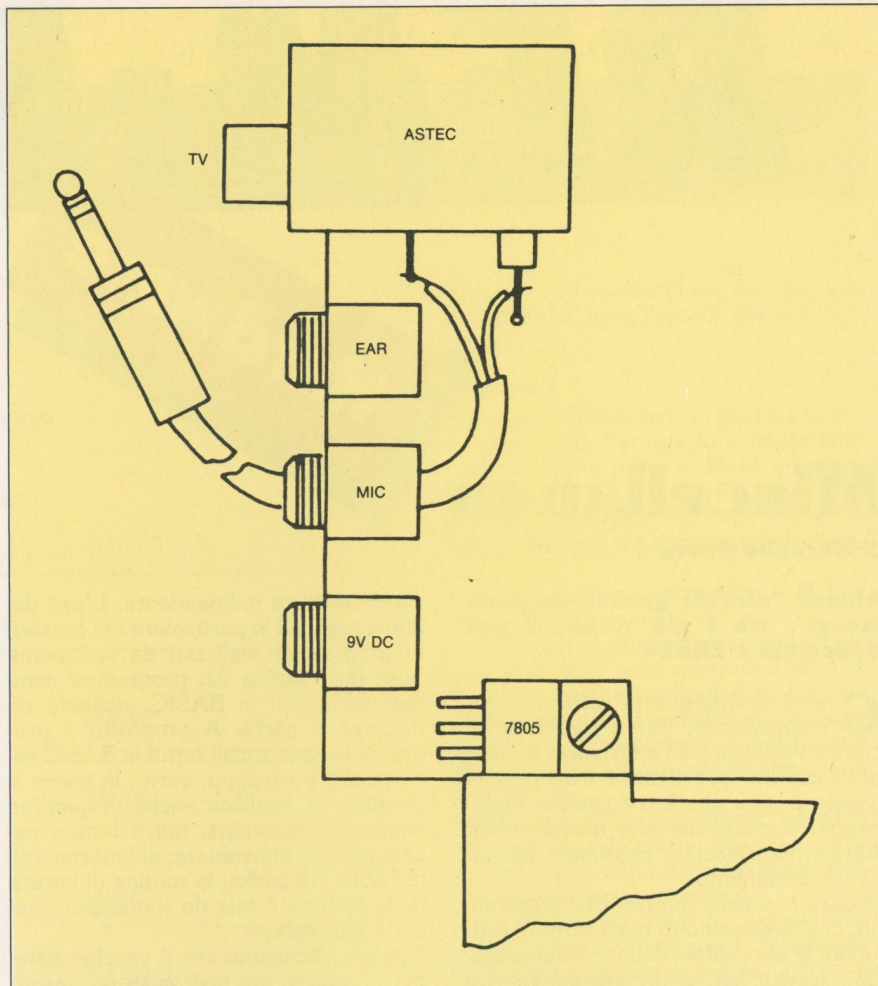


fig. 2: collegamenti all'interno dello ZX81 per la modifica alla presa MIC

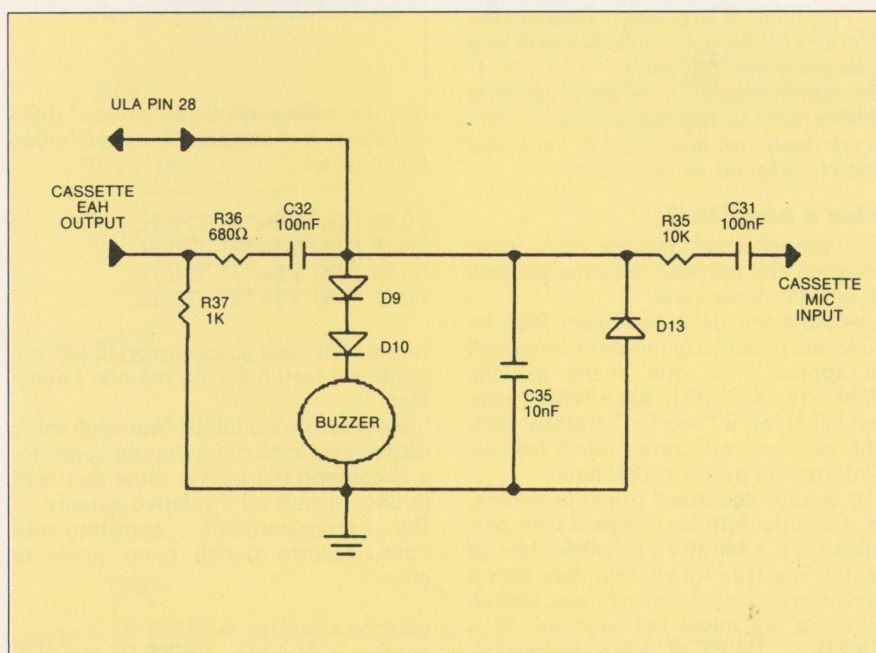


fig. 3: schema elettrico dei circuiti EAR e MIC dello Spectrum

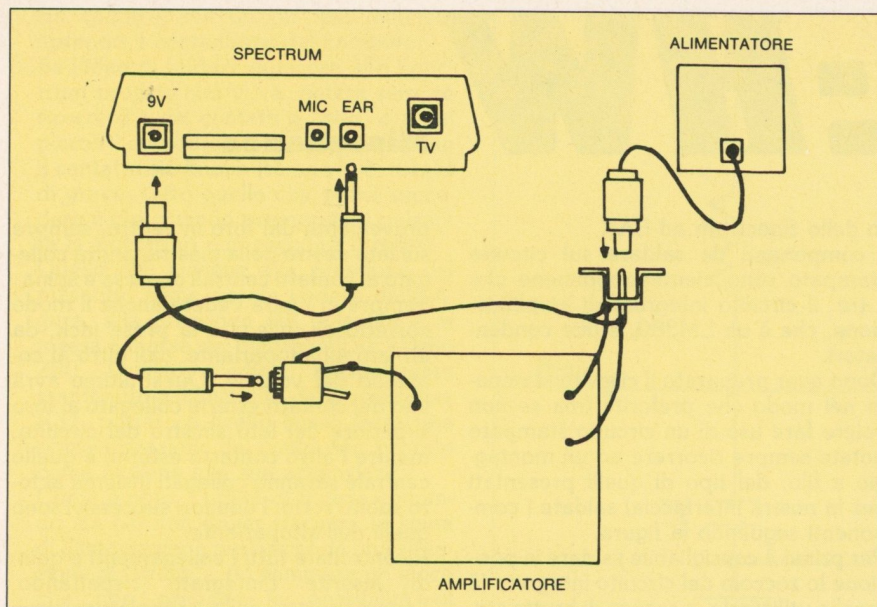


fig. 4: amplificatore per il BEEP dello Spectrum: schema delle interconnessioni

La durata di questo segnale è pari al tempo che normalmente trascorre, in religioso silenzio, fra il NEWLINE e l'inizio della trasmissione dei dati.

Ecco cosa fare:

- aprite la "carrozzeria" dello ZX81, togliendo le solite viti sul fondo;
- dopo aver fatto scivolare in avanti la parte superiore, con le solite precauzioni per evitare il distacco dei connettori di tastiera, vedrete sulla sinistra del circuito la scatola metallica del modulatore;

- questa volta i due contatti di ingresso "guardano" verso di voi. A sinistra trovate il positivo, a destra il negativo;
- solite saldature ad un cavo schermato (calza con il negativo, conduttore centrale con il positivo) che questa volta, volendo, potete fare uscire dal contenitore attraverso la presa MIC, che d'ora in avanti non verrà più usata, essendo il vecchio cavetto MIC sostituito da quello appena installato.

Questo "passaggio" non danneggia comunque minimamente la presa stessa, che potrà essere rimessa in funzione in qualsiasi momento;

- all'estremità opposta del cavo collegherete una spina tipo jack da 3,5 mm. Al positivo farete corrispondere il contatto anteriore, al negativo quello posteriore.

Il vantaggio dell'esistenza di una nota iniziale nella registrazione di dati sta nel fatto che la regolazione automatica di volume del registratore (tutti i registratori portatili a cassette ne fanno uso) si porta al giusto livello prima dell'inizio dei dati. Viene così eliminato il rischio di una cattiva registrazione, soprattutto dei primi byte, che sono i più importanti.

Il vantaggio dell'esistenza di una nota iniziale nella registrazione di dati sta nel fatto che la regolazione automatica di volume del registratore (tutti i registratori portatili a cassette ne fanno uso) si porta al giusto livello prima dell'inizio dei dati. Viene così eliminato il rischio di una cattiva registrazione, soprattutto dei primi byte, che sono i più importanti.

Per un LOAD sicuro...

Per rimanere in argomento, ma passando allo Spectrum, coloro che avessero

avuto dei problemi in fase di SAVE o LOAD possono provare ad usare la presa "sbagliata" per l'una o l'altra operazione; vale a dire: EAR per la fase di SAVE e/o MIC per quella di LOAD. Le due prese sono infatti elettricamente in comune per quanto riguarda il segnale, come vedete in figura. Ciò che varia dall'una all'altra è solo l'impedenza, molto alta per la presa MIC, piuttosto bassa per quella EAR. Vale perciò la pena di tentare...

È troppo debole quel BEEP?

A proposito di presa EAR: ormai (quasi) tutti saranno a conoscenza del fatto che il BEEP, ossia l'emissione sonora dello Spectrum, non è presente solo a livello del piccolo altoparlante, ma anche sulle due prese MIC ed EAR. La presa EAR in particolare, data la bassa impedenza, è adattissima al prelievo del segnale sonoro per una sua amplificazione.

Chi utilizza un monitor con ingresso audio può semplicemente collegare la presa EAR, con apposito cavo, all'ingresso

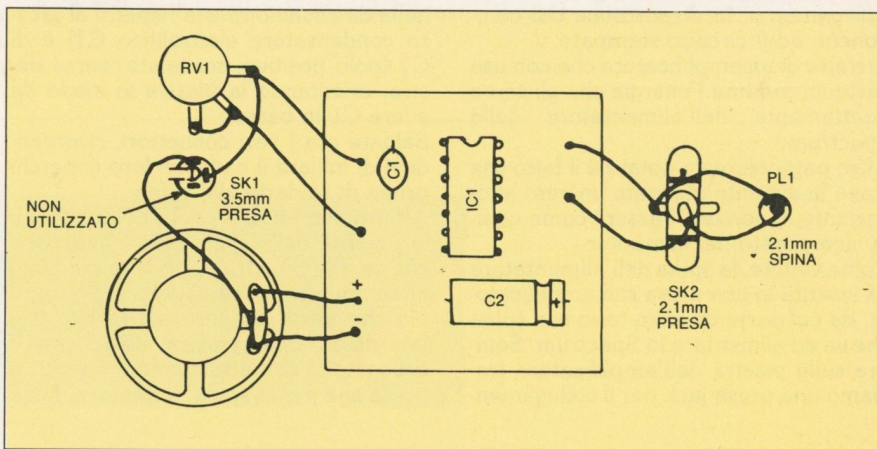


fig. 5: amplificatore per il BEEP dello Spectrum: disposizione dei componenti e dei collegamenti

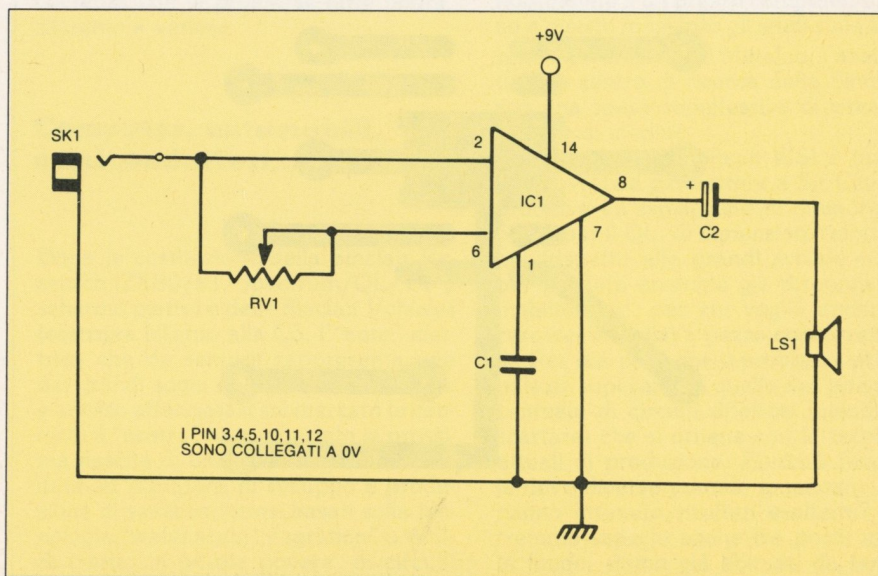


fig. 6: amplificatore per il BEEP dello Spectrum: schema elettrico

REM:HW

Miscellanea

audio del monitor (attenzione alla polarità: il positivo della presa è il contatto centrale in fondo, il negativo quello anulare).

Anche coloro che dispongono di un televisore provvisto di presa SCART, che magari utilizzano già per la parte video, possono sfruttare l'amplificatore interno al televisore. Basta collegare il positivo proveniente dalla presa EAR al contatto 6 della presa, ed il negativo al 4 o al 5.

I non possessori di monitor o di televisore con presa apposita non si disperino: ecco un progettino molto semplice di amplificatore, sufficiente a spaccare i timpani dei vicini con esplosioni galattiche e musicchette ossessive.

Nelle varie figure trovate gli schemi di collegamento, la disposizione dei componenti ed il circuito stampato.

Si tratta di un amplificatore che non usa batterie, ma trae l'energia che gli serve direttamente dall'alimentatore dello Spectrum.

Altro particolare da notare è il fatto che viene finalmente utilizzato un vero altoparlante, e non un "buzzer" come quello incorporato nel computer.

Come vedete, la spina dell'alimentatore va inserita in una presa sull'amplificatore, da cui parte un altro cavo con spina che va ad alimentare lo Spectrum. Sempre sulla piastra dell'amplificatore troviamo una presa jack per il collegamen-

to dello Spectrum ad EAR.

I componenti da saldare sul circuito stampato sono nientepopodimeno che ...tre: il circuito integrato di amplificazione, che è un LM380, e due condensatori.

Dopo aver preparato il circuito stampato nel modo che preferite (ma se non volete fare uso di un circuito stampato potete sempre ricorrere ad un montaggio a filo, del tipo di quelli presentati per la nostra interfaccia) saldate i componenti seguendo la figura.

Per primo è consigliabile saldare in posizione lo zoccolo del circuito integrato (è meglio utilizzarlo, a scanso di brutte sorprese). Quindi sarà la volta dei due condensatori. Fate attenzione all'orientamento dello zoccolo (la scanalatura va nella direzione opposta rispetto al grosso condensatore elettrolitico C1) e di C1 (polo positivo, scanalato, verso destra, orientando la piastra in modo da avere C1 in basso).

Saldate ora i vari connettori, ricordandovi di infilare il cavo nei loro coperchi prima di saldarlo alle spine.

Attenzione, MOLTA ATTENZIONE alla polarità della spina di alimentazione che va allo Spectrum, pena gravi danni al computer. Come vedete in figura, il filo che parte dal foro più in alto, sul lato destro della piastra, dovrà essere collegato al contatto esterno, sia per la presa che per la spina. Viceversa, il filo

proveniente dal foro in centro, sempre sul lato destro della piastra, andrà collegato ai contatti centrali di presa e spina. Sempre in figura vedete anche il modo corretto di collegare la presa jack, da un lato all'altoparlante, dall'altro al comando del volume. Quest'ultimo avrà uno dei contatti esterni collegato al foro superiore del lato sinistro del circuito, mentre l'altro contatto esterno e quello centrale saranno collegati insieme al foro subito sotto. I due fori successivi sono quelli dell'altoparlante.

Ricontrollate tutti i collegamenti e quindi inserite l'integrato rispettando, l'orientamento della scanalatura, presente sia su di esso che sullo zoccolo. Dopo un'ultima occhiata di controllo collegate tutto e collaudate l'amplificatore con un bel

BEEP 9,0

abbastanza prolungato da consentirvi di regolare il volume (è consigliabile procurarsi una manopola da inserire sull'albero del comando, come pure fissare il comando a una base solida di legno o cartone). Cosa ne dite, eh?

I componenti sono:

IC 1 LM380

C1 condensatore da 0,1 microfarad ceramico

C2 condensatore elettrolitico da 470 microfarad, 16 volt lavoro

VR1 potenziometro logaritmico da 22K ohm

SK1 presa jack da 3,5 mm

SK2 presa coassiale di alimentazione da 2,1 mm

PL1 spina coassiale di alimentazione da 2,1 mm

Oltre a questo materiale, è necessario un saldatore, dello stagno, filo sottile per i collegamenti, cavo bipolare per il collegamento di alimentazione e una manopola per il volume.

È consigliabile infine portare con sé l'alimentatore, per utilizzarne la spina come campione nell'acquisto di SK1 ed SK2.

A tutto reset

Per concludere, vediamo come dare allo Spectrum qualcosa del suo fratello Plus: un tasto di reset.

Tutto ciò che vi serve è un pulsante, che collegherete fra il contatto 0V ed il contatto RESET del connettore posteriore, che sono rispettivamente il ventesimo ed il sesto partendo da destra, sulla faccia inferiore del connettore.

Per non bloccare il connettore posteriore le saldature vanno fatte non proprio

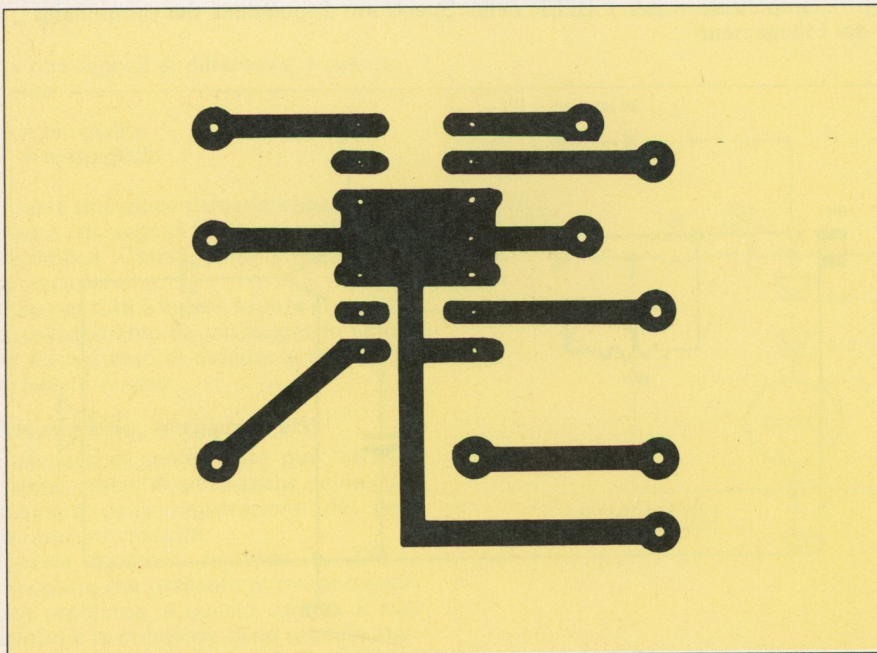


fig.7: amplificatore per il BEEP dello Spectrum: circuito stampato, lato rame

sui contatti, ma un po' più indietro, aprendo il contenitore del computer. Se l'idea di saldare qualcosa allo Spectrum proprio non vi va, potete sempre ricorrere a dei contatti a "clip" o a dei piccoli "coccodrilli": l'essenziale è che il contatto sia buono (se non lo è, niente di grave: tutto quello che potrà succedere è che quando premerete il pulsante lo Spectrum non eseguirà il reset). Quello che avviene alla pressione del pulsante, come potete vedere dalla figura, alla pressione del pulsante è la messa a massa del piedino RESET dello Z80. Questo piedino, normalmente mantenuto a livello alto dalla resistenza R31, portato a livello basso forza una reinizializzazione del computer, proprio come avviene alla sua accensione. Come è noto, l'utilità di questo comando sta nel non dover staccare la spina di alimentazione per uscire da una situazione di "crash", od eliminare un gioco che non prevede la possibilità di BREAK. In questo modo si assicura anche una vita maggiore alla presa 9V DC.

Per questa volta è tutto; arrivederci!

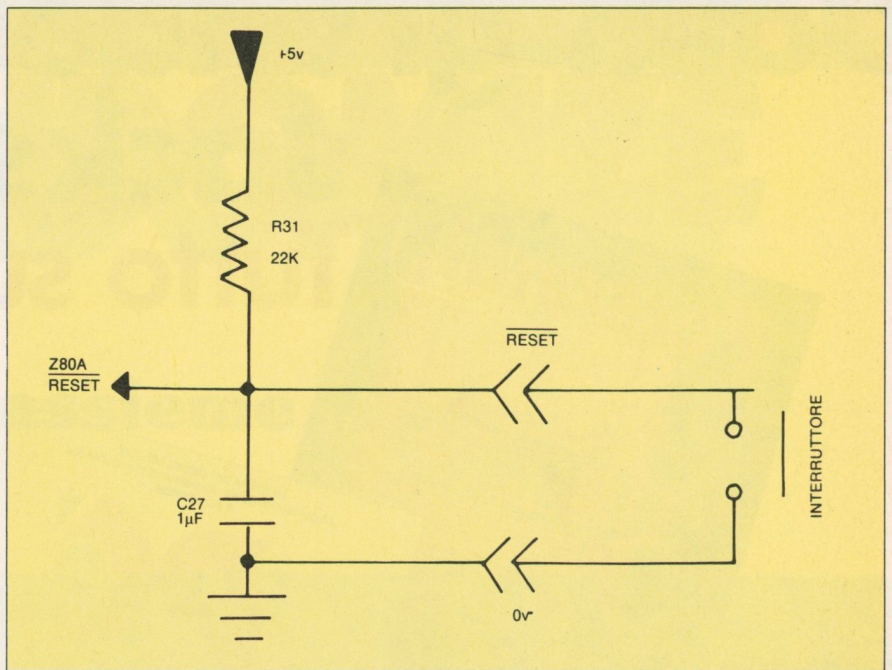


fig.8: Schema di collegamento per il tasto di reset



GBC e Sinclair camminano ancora insieme

La GBC Italiana ha rinnovato l'accordo già in vigore con la Sinclair per la distribuzione in esclusiva di Spectrum e QL. Il programma di collaborazione tra le due società prevede anche l'apertura in Italia di un ufficio di rappresentanza Sinclair, con compiti di supervisione e promozione del software, soprattutto educativo. La notizia è certo da accogliere con soddisfazione: ci chiediamo però quale sarà il ruolo, nell'ambito dell'accordo, della già da tempo annunciata Sinclair Italia, che non è ancora funzionante a pieno ritmo nonostante le prime promettenti notizie ormai risalenti a sei mesi fa. Anche degli annunciati progetti di italianizzazione del QL si sa ben poco; nel frattempo il mercato per questo computer è quasi semiclandestino, generando una diffidenza, sia pure ingiustificata, che corre il rischio di compromettere l'auspicabile lancio pubblici-

tario in grande stile, che dovrà per forza prima o poi verificarsi, se non altro per coerenza con le promesse fatte finora. Staremo a vedere.

Computer, automobili, semiconduttori,...

Dopo la costituzione della Sinclair Research (ZX80/81/Spectrum/QL/TV a schermo piatto) e della Sinclair Vehicles (costruita intorno alla C5, l'"auto" elettrica che da sempre rappresenta uno dei grandi sogni del dinamico Sir Clive e adesso affacciatasi sul mercato britannico) il "nostro" ha annunciato la prossima nascita di una nuova società, destinata ad occuparsi di sviluppo e produzione di semiconduttori basati sulla tecnologia "wafer scale integration" o WSI. Si tratta, in parole povere, di circuiti dalle notevoli prestazioni, che permettono di disporre di memorie di massa

di grande capacità - dai 512K in su - con tempi di accesso alle informazioni ridottissime a un prezzo competitivo. Fino a questo momento gli studi sulla WSI sono stati condotti al Metalab, l'avanzatissimo centro di ricerca della Sinclair, con una spesa complessiva di circa un milione di sterline.

Per inciso, la tecnologia WSI è quella che dovrebbe permettere a Sir Clive di produrre un'espansione di memoria di 512K per il QL, di dimensioni ridottissime (rispetto alle grandi schede usate per prodotti analoghi già disponibili in Inghilterra e, per chi voglia andarli a cercare, in Italia) a pezzo concorrenziale. Per ora pare che il problema principale da superare sia quello dell'alta percentuale di circuiti difettosi (quindi da scartare) che si ottiene con le tecniche attuali di produzione. Riuscirà Sinclair là dove diverse aziende americane non hanno ottenuto risultati esaltanti? Potremmo saperlo anche tra pochi mesi: in fondo, siamo già abituati da tempo alle sue sorprese, e non ci stupiremmo di constatarne una ennesima...

Dalla grande edicola Jackson

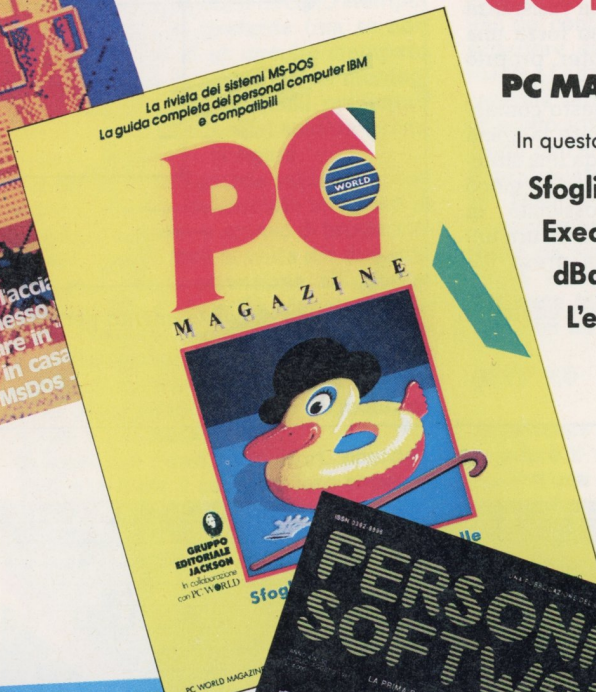
Tutto sul personal computer



PERSONAL O

In questo numero:

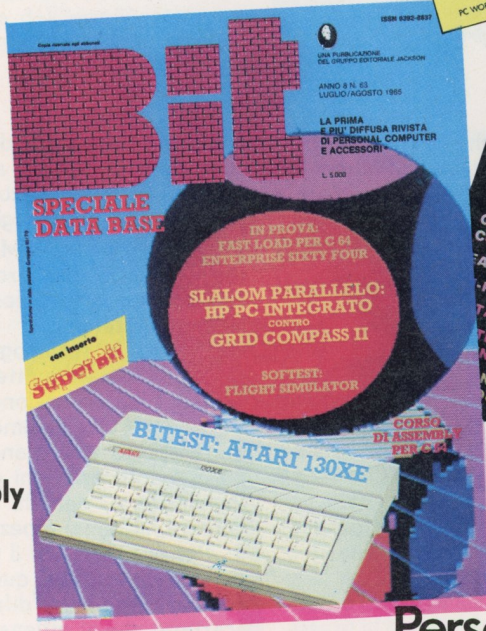
**Un video per l'acciaio
Il problema del commesso viaggiatore
Programmare in "C"
La banca in casa
Assemblatori MsDos - Pcos**



PC MAGAZINE

In questo numero:

**Sfogliando le Pagine Gialle
Execuision
dBase III
L'esperienza giapponese**



BIT

In questo numero:

**Bitest:
Atari 130XE
Slalom parallelo:
HP PC Integrato
contro
Grid Compass II
Speciale:
Data-base
Corso di Assembly
per C64**



PERSONAL SOFTWARE

In questo numero:

**Easy-Video per C64
Geo-Race per Spectrum
Data-base per C16
Disegnatore per MSX**



Personal-O/PC/Bit/Personal Software
sono pubblicazioni firmate:

GRUPPO EDITORIALE JACKSON

via Rosellini, 12-20124 Milano

RANDOMIZE

L'Assembly assieme

di **Marcello Spero**
Quarta parte

Ovverossia come eseguire operazioni logiche in linguaggio macchina.

Nel sole torrido dell'estate eccoci a parla e ancora una volta di linguaggio macchina. Chi ha seguito questo piccolo corso, puntata dopo puntata, dovrebbe ormai essere in grado di creare semplici programmi in Assembly, nonché di interpretare quelli scritti da altri.

Ci sono ancora, comunque, alcuni aspetti importanti del linguaggio macchina che vanno esaminati, prima di poter dire concluso un corso, anche se piccolo ed elementare come il nostro. L'argomento più importante lo toccheremo proprio questa volta; si tratta delle operazioni logiche. Verranno quindi l'uso delle subroutine e della catasta, le operazioni di scorrimento e gli spostamenti di blocchi. Un accenno all'uso dei registri indice concluderà questa "prima sessione" sul linguaggio macchina di ZX81 e Spectrum.

Le nozioni e le tecniche più complesse formeranno invece l'oggetto di una prossima serie, impostata ad argomenti separati, invece che a corso. Per ora cercheremo piuttosto di mettere a buon uso quanto abbiamo visto, creando routine utili e spettacolari.

A questo proposito, al termine di questa puntata troverete una sorpresa...

Stavamo dicendo...

La scorsa volta ci eravamo lasciati con la promessa di una routine che facesse uso del nuovo sistema di esecuzione. Questo sistema consiste, come abbiamo visto, nell'uso particolare dell'istruzione BASIC

DEF FN

Creando una linea tipo

10 DEF FN a(x,y,z) =USR...

al momento del suo richiamo da parte di una



RANDOMIZE FN a(i,j,k)

(o PRINT, LET r=, o quello che volete)

avremo l'esecuzione della routine indicata dalla USR, esattamente come se avessimo usato

RANDOMIZE USR...

ma in più potremo accedere, dall'interno della routine stessa, ai valori dei parametri della funzione medesima; nel nostro caso i valori di i,j,k. Ciò è possibile perché i valori dei parametri vengono materialmente inseriti

all'interno della DEF FN, in posizioni ben definite, e la variabile di sistema DEFADD ne indica la posizione.

Per essere più precisi, la situazione è quella rappresentata in figura 1. È chiaro quindi che per recuperare numeri interi e minori di 256, che sono la stragrande maggioranza dei dati utilizzati in routine in l/m, occorrerà avanzare, dalla posizione indicata da DEFADD, una prima volta di 4, e quindi di 8 per ogni successivo dato da recuperare.

Il metodo utilizzato normalmente per questa operazione è

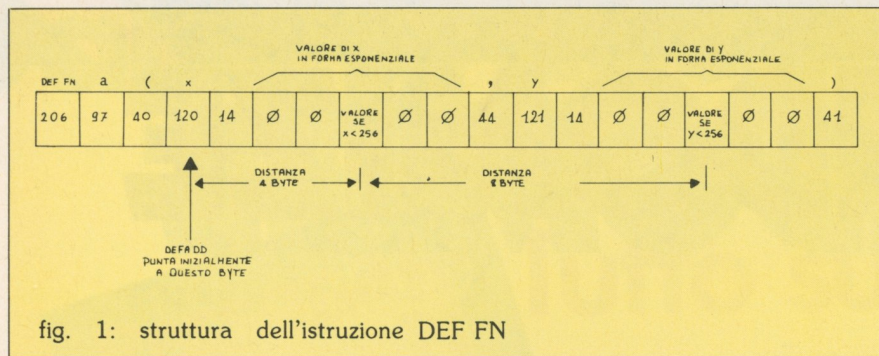


fig. 1: struttura dell'istruzione DEF FN

ld hl, (DEFADD)
 ld de, 4
 add hl, de
 ld <registro>, (hl) (valore del primo dato)

add hl, de
 add hl, de
 ld <registro>, (hl) (valore del secondo dato: un punto ogni riga, in verticale, quindi ingombro di tre righe)

È evidente una volta di più in questa routine l'utilità di disporre di una coppia come hl, che può svolgere le funzioni di registro indice e di accumulatore a 16 bit.

Il metodo usato è quello dell'indirizzamento doppiamente indiretto. Carichiamo infatti in hl il contenuto dell'indirizzo DEFADD (23563). Quindi lo incrementiamo a passi di 4 (il valore di de) e carichiamo in un registro il contenuto dell'indirizzo da esso indicato. In pratica abbiamo ottenuto qualcosa che era contenuto in qualcosa che era contenuta in DEFADD. Un po' contorto, non vi sembra?

Ed ecco un'applicazione, piuttosto semplice. Si tratta di una routine per lo Spectrum che, dati riga, colonna e valore-attributi, modifica di conseguenza il carattere indicato.

La routine,

```

ld hl, (DEFADD) 42 11 92
ld de, 4         17 4  0
add hl, de      25
ld b, (hl)      70
add hl, de      25
add hl, de      25
ld c, (hl)      78
add hl, de      25
add hl, de      25
ld a, (hl)      126
ld hl, 22528    33 0  88
ld d, 0         22 0
ld e, c         89
inc b           4
CICLO add hl, de 25
ld e, 32       30 32
djnz CICLO    16 251
ld (hl), a    119
ret           201
  
```

si limita sostanzialmente al calcolo del-

l'indirizzo desiderato, partendo dai valori di riga e colonna, e quindi all'introduzione del valore ricevuto in questo indirizzo. Il suo funzionamento è previsto in unione a una linea

DEF FN a(x, y, z) = USR <indirizzo>

in cui il primo parametro rappresenta la riga, il secondo la colonna ed il terzo il valore-attributi.

Facendo eseguire la routine con RANDOMIZE FN a(R, C, A)

dove r, c ed a sono costanti o variabili che rappresentano i vari dati nell'ordine descritto, al termine del primo blocco di operazioni avremo:

in b il numero di riga
 in c il numero di colonna
 in a il valore-attributi

La meccanica per calcolare l'indirizzo del carattere desiderato, a partire dal numero di riga e di colonna, si riduce sostanzialmente a sommare all'indirizzo di base dell'area attributi il valore di colonna, e quindi aumentare il risultato di

32*R

dove R è il numero di riga. Dovendo eseguire delle somme a 16 bit (l'indirizzo di base è 22528, evidentemente troppo grande per un registro solo), la scelta obbligata è hl. L'altro addendo potrà essere bc o de. Al momento del recupero del numero di colonna la scelta è obbligata; l'unico registro libero è infatti c. D'altronde questo non è un problema, visto che il dato corrispondente può variare fra 0 e 31, ed è quindi perfettamente a suo agio in un unico registro. Per sommare questo valore ad hl, invece, occorre utilizzare per forza una coppia, visto che l'operazione

add hl, <registro>

non esiste. Si tratta ora di scegliere: liberare b e usare bc, o trasferire il dato da c ad e e usare de?

La scelta cade sulla seconda ipotesi, visto che, terminata la fase di caricamento dati, il valore contenuto in de non ci serve più. D'altra parte ci fa comodo avere il numero di riga in b, poiché è proprio con b che funziona l'operazione

DJNZ

molto adatta ad eseguire il ciclo di somme ripetute; necessario ad eseguire l'operazione

hl = hl + 32*R

Ecco quindi che il valore contenuto in c viene ricomposto in de con le due operazioni

ld d, 0
 ld e, c

A questo punto un modo di procedere potrebbe essere

add hl, de

e quindi

```

ld de, 32
CICLO add h2, de
djnz CICLO
  
```

per aggiungere il valore di riga. Questo sistema ci pone però lo stesso problema incontrato nella routine per lo ZX 81, vista la scorsa volta. Nel caso di b, cioè numero di riga, uguale a 0, il ciclo si ripete 256 volte. In quella routine si era ovviato all'inconveniente con

ld a, b
 cp 0
 jr z,...

che in caso di b=0 saltava completamente il ciclo.

Qui la soluzione scelta è, se si vuole, un tantino più "raffinata":

```

CICLO inc b
      add hl, de
      ld e, 32
      djnz CICLO
  
```

Viene cioè inclusa nel ciclo anche la prima somma, quella in cui de contiene il numero di colonna. Per questo, b deve essere incrementata di 1. Non ponendo il valore di b essere, per ovvi motivi, inferiore a 0, il suo valore minimo dopo l'incremento sarà 1. Con un valore di contatore uguale a 1 il ciclo verrà percorso una sola volta. L'unica somma che avverrà sarà perciò quella del numero di colonna; a de verrà assegnato il valore 32, che però non sarà utilizzato.

Unico svantaggio di questo sistema è il

fatto che ad ogni esecuzione del ciclo viene ripetuta l'operazione

ld e,32

(notate come sia sufficiente caricare il valore nel registro "basso" della coppia, visto che il valore 32 è minore di 256 e il registro d contiene già 0) inutili dopo la prima volta. Si tratta comunque di uno svantaggio minimo in termini di tempo di esecuzione, mentre il vantaggio per quanto riguarda la chiarezza e compattezza della routine è notevole.

Infine avviene il caricamento del valore di a nell'indirizzo indicato da hl, eseguendo così la modifica agli attributi del carattere voluto.

Dopo aver caricato la routine, provatela con

```
10 DEF FN a(x, y, z) = USR <indirizzo>
```

```
RANDOMIZE FN a(10, 10, 248)
```

che farà lampeggiare il carattere 10, 10 fra il bianco ad alta luminosità e il nero.

Come abbiamo già avuto modo di osservare la scorsa volta, il vantaggio dell'uso di FN e DEF FN con il linguaggio macchina non rende semplicemente possibile il trasferimento di un numero qualsiasi di parametri alla routine; è persino possibile trasferire il valore di variabili, il risultato di operazioni algebriche o logiche e di altre funzioni. Impostando, ad esempio, l'esecuzione di una routine con

```
DEF FN a(x, y, z) = USR...
```

le forme

```
RANDOMIZE FN a(1, 2, 3)
RANDOMIZE FN a(f, h, 5*3)
RANDOMIZE FN a((g=6), SQR 4, 2~4)
RANDOMIZE FN a(FN b(d, 3*4), (2 AND (r > 4 OR s=9)), VAL INKEY$)
```

sono tutte possibili. L'unico vincolo da tener presente è la necessità che i risultati dei calcoli siano numeri interi.

Passiamo ad altro

Riprendiamo ora ad estendere le nostre conoscenze in fatto di linguaggio macchina, entrando in un campo che finora abbiamo tralasciato: le operazioni logiche.

Noi tutti sappiamo che in BASIC è possibile effettuare operazioni logiche. Si tratta di confronti, che producono un certo risultato dipendente dal tipo di operatore usato (gli operatori logici in BASIC sono <, >, <=, >=, < >, =, AND, OR, NOT).

Gli operandi, cioè i valori oggetto del confronto, non vengono modificati.

Ad esempio:

se a=2 e b=5

```
(a > b)=0
(a < b)=1
(a <= b)=1
(a < > b)=1
(a AND b)=a
(a OR b)=1
(NOT b)=0
```

se a=0 e b=6

```
(a AND b)= a cioè 0
(a OR b)=1
(b OR a)=b
(b AND a)=0
(NOT a)=1
```

per lo Spectrum. Più in generale,

se a=0 e b=4

```
(a AND b)=FALSO
(b AND a)=FALSO
(a OR b)=VERO
(b OR a)=VERO
```

In linguaggio macchina le cose vanno diversamente. Per effettuare dei confronti "non distruttivi" nei confronti degli operandi bisogna ricorrere all'operazione

```
cp N
```

```
o
```

```
cp <registro>
```

che può sostituire validamente gli operatori >, <, >=, <=, < >, =. Le combinazioni tipo

```
a > b AND c < d
```

si realizzano con salti condizionati:

```
ld a, N
cp b
jr c, FALSO
ld a, c
cp d
jr nc, FALSO
jr VERO
```

Infatti, se $a < b$ la sottrazione di b da a provocherà un prestito, e quindi "alzerà" il flag di carry. Viceversa, se c (che abbiamo trasferito in a per rendere possibile il confronto) $> d$, il flag di carry sarà a zero, non essendoci prestito nella sottrazione $c-d$.

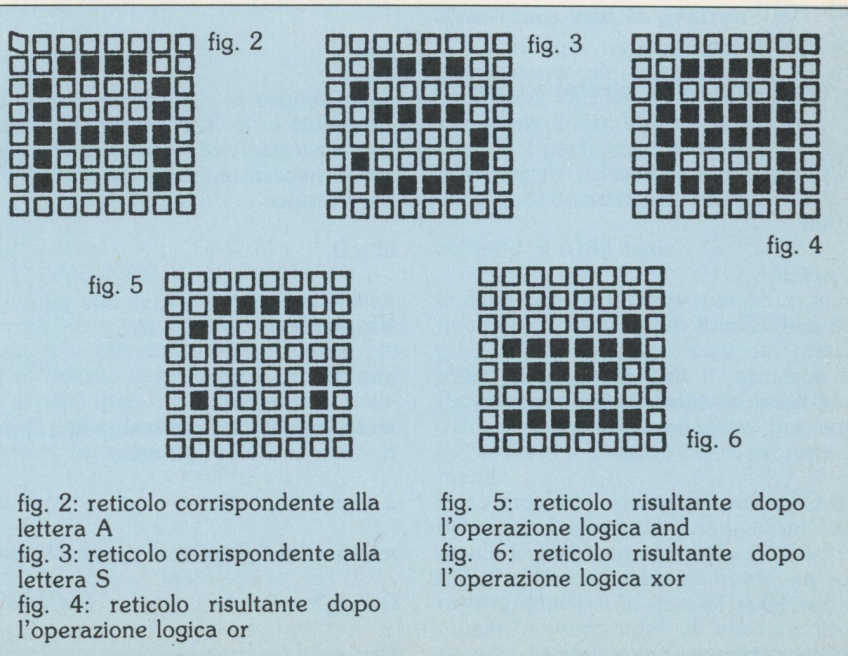
In questo modo tutti i valori oggetto dei confronti escono "sani e salvi" dall'operazione.

Esistono poi le "operazioni logiche":

```
and N
and <registro>
or N
or <registro>
xor N
xor <registro>
```

Si tratta di operazioni che agiscono non sul valore degli operandi nel suo complesso, ma sui singoli bit. Il risultato di

RANDOMIZE



queste operazioni, inoltre, viene posto in a, causando la perdita di uno dei due operandi.

Per comprenderne il funzionamento esaminiamo innanzitutto le "tabelle di verità" degli operatori logici corrispondenti.

a	b	a and b
0	0	0
0	1	0
1	0	0
1	1	1

a	b	a or b
0	0	0
0	1	1
1	0	1
1	1	1

a	b	a xor b
0	0	0
0	1	1
1	0	1
1	1	0

Gli operatori and (in italiano "prodotto logico" o "congiunzione") ed or (in italiano "somma logica" o "disgiunzione inclusiva") li conosciamo bene, in quanto presenti nel BASIC.

Qui, però, a e b non sono numeri, ma singoli bit.

In altre parole, se in BASIC

10 AND 12 = 10

per lo Spectrum, e più in generale

10 AND 12 = VERO

in linguaggio macchina i due numeri vanno considerati nella loro forma binaria:

10=1010
12=1100

L'operazione logica avverrà su ciascun bit:

1010
1100
1000

e perciò

10 and 12 = 8

In sostanza, il risultato avrà un bit a uno nelle posizioni in cui entrambi gli operandi avevano un bit a uno. E ancora:

10 OR 12 = 1

in BASIC, ma

10 or 12 = 15

in linguaggio macchina. Infatti:

10 = 1010
12 = 1100
1110 = 14

Dove almeno uno degli operandi aveva un bit a uno, nel risultato ci sarà un uno.

XOR, l'OR esclusivo

Vediamo adesso l'ultimo operatore, meno familiare. La sua tabella di verità, apparentemente complicata, può essere riassunta nella definizione:

diverso = VERO
uguale = FALSO

Infatti, sia la combinazione

a=1 b=1

che

a=0 b=0

danno come risultato zero, o FALSO che dir si voglia. Vediamo ora un esempio in linguaggio macchina:

12 xor 5 = 9

infatti:

14 = 1100
7 = 0101
1001 = 9

Dove i bit dei due operandi erano diversi troviamo un uno, dove erano uguali uno zero.

Ma... e in pratica?

Ci arriviamo, ci arriviamo...

Vedrete che non mancherete di stupirvi per la quantità di cose che si possono fare con queste operazioni.

Vediamo qualche caso limite:

and 0

darà sempre zero, qualsiasi valore sia contenuto in a. Si tratta di un sistema per azzerare l'accumulatore, che però non è necessariamente da preferire al più prosaico

ld a,0

poiché occupa anch'esso due byte. Viceversa,

and 255

lascia immutato il contenuto di a. Infatti, dove in a c'era un uno,

1 and 1 = 1

mentre dove c'era uno zero

0 and 1 = 0

Per or le cose si ribaltano:

or 255

darà invariabilmente 255 come risultato. Viceversa,

or 0

lascia immutato il contenuto di a.

Le operazioni "simmetriche", vale a dire

and a

e

or a

lasciano immutato il contenuto dell'accumulatore.

Le cose non stanno assolutamente così per xor.

xor a

infatti, darà sempre e solo zero, poiché i bit nella stessa posizione saranno tutti uguali. Si tratta quindi di un modo rapido e breve (occupa un solo byte in memoria) per azzerare l'accumulatore. Interessanti sono anche i due casi

xor 0

che lascia immutato il contenuto di a, ed

xor 255

che esegue il complemento a due dei bit che compongono il valore. Quest'ultimo caso merita di essere svolto per esteso; prendiamo, ad esempio,

a=38

38 = 00100110
255 = 11111111
11011001 = 217

che corrisponde a

255-38

Infatti il complemento a due, in senso binario, di un numero equivale al suo complemento a 255 in senso decimale.

In generale, si usa l'operazione or quando si ha la necessità di portare a 1 il valore di uno o più bit all'interno di a.

I bit in corrispondenza degli zeri saranno rispettati; quelli in corrispondenza degli uno passeranno a 1.

Viceversa, si usa and per portare a zero uno o più bit di a. In questo caso saranno i bit in corrispondenza degli uno ad essere rispettati, mentre quelli in corrispondenza degli zeri passeranno inesorabilmente a zero.

Questi due operatori possono servire

anche per una specie di lavoro di "colage". Supponiamo di voler comporre in a un numero che abbia i primi quattro bit di b e i secondi quattro di c. Le operazioni di fare sono:

```
ld a,15
and b
ld b,a
ld a,240
and c
or b
```

che equivalgono a:

```
ld a,b
and 15
ld b,a
ld a,c
and 240
or b
```

Vediamo un po' il meccanismo. Per capirci meglio, supponiamo che

```
b = 10110101 (181)
c = 01101011 (107)
```

Noi vogliamo che il valore finale in a sia

```
01100101 (101)
```

ossia un numero composto dai quattro bit di ordine superiore di c e dai quattro di ordine inferiore di b.

La prima operazione è, praticamente,

```
b and 15
```

cioè

```
10110101 (181)
00001111 (15)
00000101
```

Quindi si passa alla preparazione di c, che avviene con

```
c and 240
```

cioè:

```
01101011 (107)
11110000 (240)
01100000
```

A questo punto non resta che unire la due metà con una operazione di or:

```
00000101
01100000
01100101 (101)
```

Lo stesso risultato si può ottenere con:

```
ld a,240
or b
ld b,a
ld a,15
or c
and b
```

che equivale a



RANDOMIZE

```
ld a,b
or 240
ld b,a
ld a,c
or 15
and b
```

Infatti:

```
10110101 (181) or
11110000 (240)
11110101
```

```
01101011 (107) or
00001111 (15)
01101111
```

ed infine

```
11110101 and
01101111
01100101 (101)
```

Meno intuitivo è l'uso di xor.

Questo operatore può essere utilizzato per verificare quali bit di due registri, o di un registro e un dato, sono uguali. Come risultato otterremo un byte in cui i bit a zero significheranno "qui i due operandi avevano un bit uguale", mentre quelli a uno "qui i due operandi avevano un bit diverso l'uno dall'altro".

Vediamo un esempio. Sia

```
a=10110001 (177)
b=00101001 (41)
```

l'operazione

```
xor b
```

darà:

```
10110001
00101001
10011000 (152)
```

Un altro impiego utile di questa operazione è la commutazione di uno o più bit all'interno di un registro, anche senza sapere qual è il loro valore. Possiamo di voler commutare il bit 3 di a. Basterà far eseguire

```
xor 8
```

infatti la forma binaria di 8 è

```
00001000
```

Abbiamo detto che con xor i bit a zero

non provocano alcuna modifica, mentre quelli a uno complementano a due (cioè invertono) il bit corrispondente. Perciò il bit 3 di a sarà invertito, mentre gli altri saranno rispettati. Un'applicazione dell'operatore xor, presente in tutti i computer, riguarda la ben nota "regola del segno".

Tutti noi sappiamo che nel moltiplicare o dividere due numeri occorre tener conto, per determinare il segno del risultato, della regola:

```
+ xo: + = +
+ xo: - = -
- xo: + = -
- xo: - = +
```

Se, come di fatto avviene nella maggior parte dei computer, il segno di un numero viene rappresentato con lo stato di un bit, che si stabilisce

```
0 = +
1 = -
```

per ottenere il segno del risultato di una moltiplicazione o divisione basterà eseguire un xor dei segni degli operandi. Infatti

```
0 xor 0 = 0
1 xor 0 = 1
0 xor 1 = 1
1 xor 1 = 0
```

Proviamo con la grafica

C'è comunque un modo più diretto ed interessante per comprendere queste operazioni logiche, ed è osservando visivamente il loro effetto sui singoli bit. Con lo Spectrum questo è possibile. L'immagine video è infatti il risultato della visualizzazione di ben

$32 \times 24 \times 8 = 6144$ byte

in cui a ciascun bit corrisponde un puntino nero (o meglio, del colore INK), se questo è a 1, bianco (cioè del colore PAPER), se è a 0.

Provando le operazioni logiche sui byte dell'area video, saremo quindi in grado di "vedere" il risultato dei nostri esperimenti.

Per comodità manipoleremo gli otto byte che formano il primo carattere, quello in alto a sinistra per intenderci. Ciò che faremo sarà stampare un carattere qualsiasi in questa posizione, e quindi "sovrapporgli" un altro carattere, usando a turno i tre operatori logici.

In parole più esatte, eseguiremo un'operazione di or, and o xor fra ciascuno degli otto byte che compongono il carattere ed il corrispondente byte di un altro carattere.

Le operazioni da fare sono:

- stampare, con una PRINT in BASIC, il carattere desiderato in posizione 0,0;
- prelevare, uno a uno, i byte di un altro carattere dal set di caratteri contenuto nella ROM dello Spectrum;
- eseguire, su ciascuna coppia corrispondente di byte, l'operazione logica desiderata;
- collocare ciascun risultato nell'opportuno indirizzo dell'area video, in modo da ricomporre con i nuovi byte modificati un carattere nella stessa posizione di quello iniziale.

Il primo punto è presto eseguito:

CLS : PRINT "A"

Per i punti successivi dobbiamo invece costruirci un'apposita routine in linguaggio macchina.

Le informazioni che ci servono sono:

- posizione del primo byte del primo carattere dell'area video:
indirizzo 16384
- distanza nell'area video fra due byte successivi di uno stesso carattere:
256 byte
- posizione, all'interno della ROM, del primo byte del primo carattere c'è lo spazio del set:
indirizzo 15616
- distanza del primo byte di ciascun carattere dal primo byte del carattere precedente:
8 byte

Decidiamo di utilizzare, come carattere da "sovrapporre", la S. Il suo CODE è 83, che sottratto a quello dello spazio, 32, ci dà una distanza dal primo carattere di

$$83-32=51 \text{ caratteri}$$

Moltiplicando questa distanza per 8 (il numero di byte fra l'iniziale di un carattere e l'iniziale del successivo) otteniamo

$$51*8=408 \text{ byte}$$

L'indirizzo del primo byte della lettera S sarà perciò

$$15616 + 408 = 16024$$

Ora possiamo scrivere la nostra routine:

```
ld hl,16384      33  0  64
ld de,16024     17 152  62
ld b,8          6   8
LOOP ld a,(de)  26
or (hl)         182
ld (hl),a      119
```

```
inc h           36
inc de         19
djnz LOOP     16 249
ret           201
```

Il suo funzionamento è molto semplice:

```
ld hl,16384
ld de,16024
ld b,8
```

- in hl l'indirizzo del primo byte dell'area video, in de quello del primo byte del carattere S, all'interno del set, in b 8 per controllare il ciclo (sono infatti 8 i byte che compongono un carattere).

```
ld a,(de)
or (hl)
ld (hl),a
```

- ciascun byte componente la S viene recuperato e caricato in a, e quindi avviene l'operazione logica (in questo caso or) con il byte corrispondente nell'area video. A questo proposito notate come non sia necessaria l'operazione preliminare

```
ld c,(hl)
```

e quindi

```
or c
```

ma sia possibile, come abbiamo già visto altre volte, usare (hl) come se fosse un registro. Altrettanto NON si potrebbe fare con (bc) o (de).

```
inc h
inc de
djnz
```

- l'indirizzo nell'area video indicato da hl, viene aumentato di 256 con l'incremento del byte più significativo della coppia. L'indirizzo all'interno del set viene invece aumentato solo di uno. Non è possibile l'incremento unitario con

```
inc e
```

poiché il valore contenuto in questa coppia potrebbe essere

$$d=X \ e=255$$

ed in tal caso l'operazione

```
inc e
```

produrrebbe il risultato

$$d=X \ e=0$$

non superiore di uno, ma inferiore di 255 al precedente. Con

```
inc de
```

si ha invece

$$d=X+1 \ e=0$$

Sostituendo al byte I+9, dove I è l'indirizzo del primo byte della routine, i valori

```
166 per and
174 per xor
182 per or
```

potrete sperimentare i risultati delle varie combinazioni.

Le figure 2, 3, 4, 5 e 6 illustrano, rispettivamente:

- la disposizione dei bit per il carattere A;
- la disposizione dei bit nel carattere S;
- il risultato dell'operazione or;
- il risultato dell'operazione and;
- il risultato dell'operazione xor.

In questo modo potrete, più agevolmente che non sullo schermo televisivo, confrontare riga per riga (cioè byte per byte) operandi e risultati. Per quanto riguarda l'operazione xor c'è da notare che si tratta della stessa che viene effettuata quando si stampa sullo schermo in modo

OVER 1

A riprova di questo, eseguendola per due volte consecutive riappare il carattere originario, cioè la A.

E per finire...

Basta con gli esempi! A partire da questa puntata iniziamo la presentazione di routine di una certa complessità ed utilità.

Prese separatamente potranno essere inserite dove vi serviranno; noi, comunque, le utilizzeremo per "costruire" un grandioso "arcade", che nulla avrà da invidiare a un prodotto commerciale.

Non essendo possibile portare avanti un progetto del genere parallelamente per Spectrum e ZX81, si è imposta una scelta. La ben più ampia diffusione, oltre che le maggiori possibilità grafiche, hanno fatto pendere la bilancia a favore dello Spectrum; gli ottantunisti, comunque, non saranno certo abbandonati: anche per loro ci sarà un "game", pur se necessariamente meno impegnativo. Torniamo adesso al nostro progetto.

Le routine relative alla struttura base del gioco compariranno in queste ultime puntate del corso; gli "accessori" e le raffinatezze le vedremo invece nelle varie puntate monografiche che seguiranno.

Per quanto riguarda il blocco che vi viene presentato questa volta, c'è da dire che molte cose al suo interno non sono ancora state spiegate.

Si tratta dell'uso dello stack, delle sub-routine e del registro indice ix, nonché lo scorrimento di registri. Le prossime puntate chiariranno questi punti, che è

stato necessario "anticipare" in questa routine per darvi qualcosa di completo e funzionante già così com'è.

Si tratta nientedimeno che di una routine per produrre il movimento di caratteri grafici in alta risoluzione. Non, cioè, di un carattere, ma di un pixel per volta.

È strutturata per poter essere facilmente utilizzata dal BASIC, con il sistema

DEF FN

appena visto.

Novità: vista l'estensione del listato, trovate riprodotto l'output dell'Assembler, per evitare il rischio di errori. Questo ha anche il vantaggio di favorire una certa familiarizzazione con gli Assembler. Per questo vedrete in futuro non solo listati prodotti con l'Assembler della DK'Tronics, usato questa volta, ma anche con assembleri di altre marche.

Vediamo di chiarire alcune cose circa questo nuovo modo di presentare l'Assembly:

- in ciascuna riga trovate, da sinistra a destra, l'indirizzo del primo byte dell'operazione, i codici esadecimali dell'operazione stessa ed i codici Assembly;

- noterete, in testa al listato, alcune linee "equ". Servono ad attribuire determinati valori a delle "etichette" (o label), che saranno usate in seguito, all'interno della routine. Questo sistema permette di avere a che fare con abbreviazioni che richiamano subito alla mente il loro significato, invece che con aridi numeri. Le etichette non definite con linee equ riguardano operazioni di

salto all'interno della routine, e perciò assumono automaticamente il loro valore, a seconda della posizione. Maggiori spiegazioni in futuro;

- alcuni valori, all'interno del listato, sono preceduti dal carattere "&". Si tratta di valori espressi in forma esadecimale;

- codici decimali, adatti ad essere caricati con il solito metodo, li trovate nel listato 2. Attenzione! Per motivi che chiariremo a partire dalla prossima volta questa routine NON È RILOCABILE, cioè deve necessariamente risiedere a partire dall'indirizzo 30000.

Pur rimandando alla prossima puntata i dettagli di funzionamento, vediamo già ora tutte le informazioni che possono esservi utili per utilizzare subito la routine.

Dopo averla caricato, preparate la consueta linea BASIC

DEF FN a(A, B, C, D)

dove:

A sarà uguale a 1, se non si desidera la cancellazione del carattere nella sua posizione precedente (perché si tratta della prima stampa di quel carattere, o perché si vuole lasciare una "scia"). 0 viceversa.

B sarà il numero d'ordine del carattere grafico che vogliamo usare: 0 per la A, 1 per la B e così via.

C sarà la coordinata x della posizione dell'angolo superiore sinistro del carattere.

D sarà la coordinata y della stessa posizione. Per comodità l'asse delle y è stato rovesciato. Le coordinate

0,0

corrispondono perciò all'angolo superiore sinistro dello schermo, mentre le coordinate

255,191

(siamo in alta risoluzione, non dimenticatelo) all'estremo inferiore destro. Come vedete dal valore di quest'ultima coppia di coordinate, sono utilizzabili anche le due righe normalmente destinate ai comandi ed ai messaggi. Naturalmente, bisognerà tener conto che un carattere occupa 8 pixel in larghezza e 8 in altezza; l'ultima coppia utile di coordinate sarà perciò

248,184

(cioè 255-7, 191-7)

Attenzione a non uscire da questi limiti, o saranno guai!

Per provare potete utilizzare il programma

```

10 DEF FN a(a,b,c,d)=USR 30000
20 LET x=0:LET y=0
30 RANDOMIZE FN a(1,0,x,y)
40 LET x=x+(INKEY$="8")-(INKEY$="5")
50 LET y=y+(INKEY$="6")-(INKEY$="7")
60 IF x>248 THEN LET x=248
70 IF y>184 THEN LET y=184
80 IF x>0 THEN LET x=0
90 IF y>0 THEN LET y=0
100 RANDOMIZE FN a(0,0,x,y)
110 PAUSE 0
120 GO TO 40

```

che vi darà un'idea delle potenzialità della routine.

list. 1: Assembly della routine grafica

equ 23728 COOR	50043 EB	ex de,hl
equ 23729 Y_COOR	50044 3A 81 5C	ld a,(CHAR_NR)
equ 23681 CHAR_NR	50047 6F	ld l,a
equ 23675 UDG	50048 26 00	ld h,0
equ 23563 DEFADD	50050 29	add hl,hl
org 50001	50051 29	add hl,hl
50001 2A 0B 5C	50052 29	add hl,hl
50004 11 04 00	50053 19	add hl,de
50007 19	50054 F5	push hl
50008 7E	50055 DD E1	pop ix
50009 A7	50057 ED 4B B0 5C	ld bc,(COORD)
50010 76	50061 CD B5 C3	call PIXEL
50011 20 07	50064 F5	push af
50013 E5	50065 47	ld b,a
50014 D5	50066 CD CE C3	call PLOT
50015 CD 76 C3	50069 DD 23	inc ix
50018 D1	50071 ED 4B B0 5C	ld bc,(COORD)
50019 E1	50075 48	ld c,b
50020 19	50076 06 07	ld b,7
50021 19	50078 0C	inc c
50022 7E	50079 C5	push bc
50023 32 81 5C	50080 CD BC C3	call DIS
50026 19	50083 C1	pop bc
50027 19	50084 0C	inc c
50028 7E	50085 F1	pop af
50029 32 B0 5C	50086 F5	push af
50032 19	50087 C5	push bc
50033 19	50088 47	ld b,a
50034 7E	50089 CD CE C3	call PLOT
50035 32 B1 5C	50092 DD 23	inc ix
50038 DD E5	50094 C1	pop bc
50040 2A 7B 5C	50095 10	djnz LOOP_M
	50097 F1	pop af

RANDOMIZE

Sabre Wulf: una giungla 16x16

di **Alberto Barbati**
e **Flavio Bruschi**

Ecco la mappa di uno fra i più celebri adventure animati per lo Spectrum, realizzata da due giovanissimi lettori.

Sabre Wulf è il secondo episodio della serie della Ultimate cominciata con Atic Atac, anche se non è in stretta relazione con questo.

Il nostro personaggio è stavolta **SABRE MAN**, un coraggioso esploratore che dovremo aiutare nelle sue ricerche. Unica difesa del nostro eroe è il suo machete, che lo accompagnerà costantemente nella sua avventura.

Per guidare **SABRE MAN** possiamo usare uno dei joystick previsti o, con l'opzione **KEYBOARD**, ricorrere ai tasti da Q a T sulla tastiera. Ricordiamo che con l'opzione **CURSOR** è possibile usare anche i tasti cursore più lo zero. Qualunque sia l'opzione di movimento selezionata, con **CAPS SHIFT** si può fermare momentaneamente il gioco (cosa utilissima per dare un'occhiata alla mappa); riprendendo **CAPS SHIFT** il gioco continua.

Lo scopo

Lo scopo di Sabre Wulf consiste nel ritrovare i quattro pezzi di un medaglione che sono sparsi per la giungla, portandoli in un tempio. La cosa sembra a prima vista facile, ma la giungla è molto vasta e piena di insidie. Vediamo in particolare i vari aspetti del gioco.

I mostri

I mostri sono i nostri principali nemici, e senza di essi il gioco sarebbe a dir poco banale. Li possiamo dividere tra mostri autorigeneranti e non.

Quelli autorigeneranti sono quelli che appaiono sullo schermo da una specie di nuvoletta, svanita la quale iniziano a scorrazzare in lungo e in largo, cercando di porre fine alla vita del nostro omino. Sono gli unici che possono essere uccisi col nostro machete. Ce ne sono di molti tipi, ma principalmente si tratta

di insetti o rettili; man mano che ritroviamo i vari pezzi del medaglione, nuove specie di animaletti entrano a far parte della schiera dei nostri nemici.

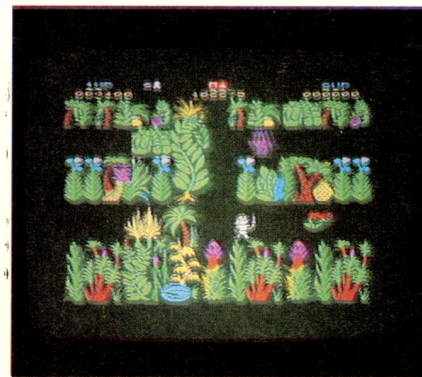
Ci sono altre insidie nella nostra giungla; ad esempio gli indigeni, che vagano per l'area di gioco e che non possono essere uccisi. Si può, al massimo, colpirli col machete, incitandoli a cambiare direzione. Ma spesso hanno la testa dura!

In giro si possono trovare anche rinoceronti o altri animali di grossa taglia, che stanno solitamente accucciati in un angolo. Basta solo toccarli per provocarne il risveglio e la furiosa carica verso il nostro beniamino, dopo di che essi riprendono il sonno interrotto. Come è logico aspettarsi, non sono molto intelli-

la fiamma. Quando il tempo di permanenza in una stanza o in un passaggio inizia a diventare troppo lungo eccola arrivare. Se non ci allontaniamo subito, ci inseguirà per incenerirci. Come per il lupo, contro la fiamma il machete non ha effetto.

Le orchidee

Lungo il percorso sono sparse delle orchidee, che, qualcuna più qualcuna meno, ci aiuteranno a terminare il gioco. Questi fiori non sono sempre utilizzabili; solitamente è visibile solo un piccolo punto e, quando uno meno se lo aspetta, il fiore sboccia e subito appassisce. Per utilizzare un'orchidea bisogna toccarla nel momento di massima fioritura. Il loro effetto dipende dal colore del



genti e basta punzecchiarli col machete per fargli cambiare idea.

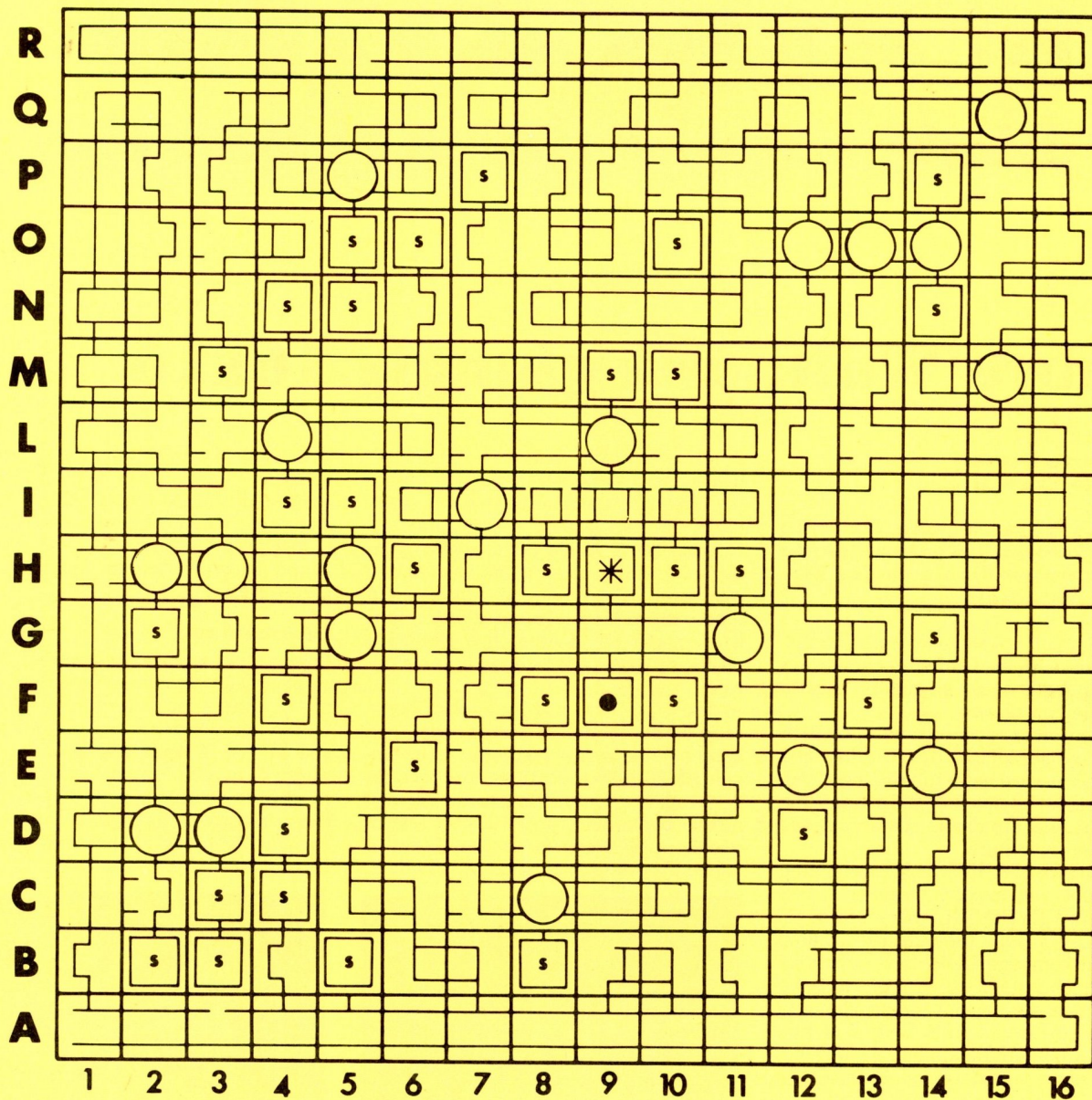
Nel tratto di caselle da A3 a A15 (vedi mappa) troviamo il più pericoloso dei mostri in circolazione: il lupo. Esso continua a vagare avanti e indietro nel succitato tratto, sperando di riuscire a catturare il nostro omino. Il solo contatto con il lupo (anche col machete) provoca la morte.

Nel tempio centrale, la nostra meta (casella M9), c'è lo stregone, che resta immobile davanti all'ingresso per impedirci di entrare. Solo dopo aver completato il medaglione diventerà inoffensivo e potremo passare.

Ma la più pericolosa delle entità malefiche che ci contrastano è senza dubbio

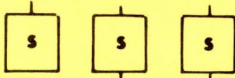



MAPPA DEL GIOCO




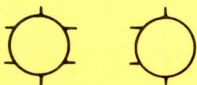
LEGENDA

— —| + ———— **PASSAGGI**

 ———— **STANZE**

 ———— **CAMERA DI PARTENZA**

 ———— **TEMPIO**

 ———— **RACCORDI O SPIAZZI**

N.B.: Ogni casella corrisponde ad una videata

RANDOMIZE

fiore ed è comunque temporaneo. L'omino diventerà del colore dell'orchidea finché dura il suo effetto. L'orchidea rossa ci rende invulnerabili a qualsiasi attacco.

Quella azzurra, sicuramente la più utile, rende il nostro omino invulnerabile e velocissimo. L'orchidea viola agisce come quella rossa, ma inverte i comandi di movimento, mentre quella bianca fa terminare immediatamente l'effetto delle altre. Quella gialla, infine, ci immobilizza per un po', ma uccide tutti i mostri visibili, tranne il lupo, gli indigeni e i rinoceronti.

I bonus

Dispersi per la foresta ci sono vari tipi di oggetti-bonus, da anelli a brillanti, da spade a sacchi di monete. Ciascuno di essi, se preso, ci dà 150 punti. Da notare il bonus omino, che, oltre ai punti, ci regala una vita (anche se non potremo mai averne più di 9).

Il medaglione

Raccogliere i quattro pezzi del medaglione è, come abbiamo già detto, la prima cosa da fare per finire il gioco. I pezzi possono essere ritrovati solo nelle stanze (vedi mappa) e mai nei passaggi. Possono essere riconosciuti dagli altri oggetti per la loro luminosità. Quando uno di questi viene preso il gioco viene interrotto momentaneamente, mostrandoci a che punto siamo della ricostruzione. Il ritrovamento viene premiato con un bonus di 7500 punti.

Consigli tattici

Una buona condotta tattica è molto importante per risolvere questo gioco; il primo consiglio che vi diamo è quello di tenere costantemente premuto il tasto che controlla il machete (O, se si usano i tasti cursore, T nell'opzione KEYBOARD). In questo modo avrete una valida copertura nel movimento orizzontale; in caso di movimento verticale tenetevi a contatto di una parete, cercando di non voltare mai le spalle ai nemici.

Non avventuratevi nelle stanze senza essere sotto l'effetto di un'orchidea o comunque non tenetevi al centro, sempre per evitare di voltare le spalle ai nemici.

Non insistete nello scontro con un indigeno; questi potrebbe avvicinarsi troppo e, nonostante il vostro machete, ucci-

dervi.

Se un rinoceronte addormentato vi ostruisce il passaggio, avvicinatevi e stuzzicatelo con la punta del machete; quindi toglietevi dalla sua traiettoria, lasciatelo andare via e proseguite il vostro cammino.

Consigli strategici

Innanzitutto, per la buona riuscita del gioco è necessario che la vostra esplorazione sia la più razionale possibile, cioè che esplorate la maggior parte della zona di gioco nel minor tempo.

Non soffermatevi in quelle stanze dove non ci sono pezzi del medaglione: soprattutto cercate di esplorare tutte le stanze secondo un ordine logico.

La mappa

La mappa è un quadrato di 16x16 caselle, e rappresenta l'intera giungla in cui è relegato SABRE MAN. Per comodità le caselle sono numerate da 1 a 16 in orizzontale e da A ad R in verticale.

Ingredienti principali della mappa sono stanze e spiazzi, collegati tra di loro per mezzo di passaggi.

Le stanze sono collegate con una o due uscite, e possono avere nel centro una capanna o un laghetto. Solo qui possiamo trovare i pezzi del medaglione. In tutta l'area di gioco ve ne sono 32, più la stanza di partenza e il tempio.

Notare che le due stanze in H3 e in H10 hanno le pareti in roccia, al contrario di tutte le altre.

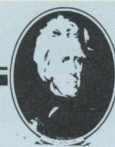
Gli spiazzi sono semplicemente dei punti in cui convergono più passaggi (solitamente 6).

Risolto il gioco, apparirà il messaggio THE NEXT ADVENTURE OF SABRE MAN WILL BE CONTINUED IN THE UNDERWULDE.

Underwulde è il titolo di un altro celebre gioco Ultimate (di cui può darsi riceverete presto nostre notizie...).

Conclusioni

Speriamo che con questi suggerimenti sarete in grado di risolvere Sabre Wulf; ci scusiamo per eventuali errori che possiamo avere commesso, invitandovi a segnalarceli. Buona partita!



Dalla grande edicola Jackson

Tutto sul personal computer

PC

L'unica rivista italiana dedicata ai sistemi MS-DOS, Personal computer IBM e compatibili.
10 numeri all'anno: L. 5.000 a numero
Abbonamento: solo L. 40.000

Personal

L'unica rivista indipendente per gli utenti dei personal computer Olivetti.
10 numeri all'anno: L. 4.000 a numero
Abbonamento: solo L. 35.000

COMPUSCUOLA

La rivista di informatica nella didattica per la scuola italiana.
9 numeri all'anno: L. 2.000 a numero
Abbonamento: solo L. 15.000

PERSONAL SOFTWARE

Aspetti e problemi del software per personal computer, programmi, giochi e sistemi operativi.
11 numeri all'anno: L. 4.000 a numero
Abbonamento: solo L. 34.000

Bit

La prima rivista europea di personal computer, software e accessori. Con test, novità, analisi del mercato...
11 numeri all'anno: L. 5.000 a numero
Abbonamento: solo L. 43.000

Quando l'informazione fa testo

In busta chiusa inviate questo coupon a:
Gruppo Editoriale Jackson
via Rosellini, 12 - 20124 Mi

Desidero ricevere GRATIS un numero della Rivista _____

(allego L. 1.000 in francobolli per contributo spese di spedizione)

Inviatemi GRATIS il Catalogo della Biblioteca JACKSON (allego L. 1.000 in francobolli per contributo spese di spedizione)

Nome _____

Cognome _____

via _____

CAP _____ Città _____

Cancellino

di **Stan Rodgers**
Trad. e adatt.
di **Lorenzo Brambille**

Bastano pochi tasti per cancellare istantaneamente le linee di programma che non vi servono!

Questo programma sfrutta una piccola routine in l/m inserita nel printer buffer, a partire dalla locazione 23300, e questo lo rende sfruttabile per Spectrum sia da 16 che da 48K.

Il programma chiama una routine di monitor situata all'indirizzo decimale a 6510 per avere il numero della prima linea da cancellare e il numero della linea successiva all'ultima di quelle da eliminare. Questi numeri di linea vengono poi passati alla routine di monitor a 6629, che li elabora ed effettua la eliminazione del blocco. La routine in sé stessa è quindi relativamente semplice e molto, molto più veloce di parecchi programmi analoghi presenti nei toolkit professionali più famosi.

Per usare il programma digitatelo e salvatelo.

Le linee 4,5,6 contengono il loader e i data necessari a creare il linguaggio macchina. Digitate GOTO 4, e il linguaggio macchina verrà collocato nella memoria dalla locazione 23300 in poi. Se compare il messaggio "Errore nel listato", avete probabilmente commesso un errore un errore nei DATA alla linea 6. Controllateli attentamente, correggete l'errore e digitate nuovamente GOTO 4. Se tutto è OK, apparirà il messaggio. "IL CANCELLINO è pron-

to!"

Adesso potete provare il programma. Digitate GOTO 1 e la routine vi chiederà il numero della linea dalla quale volete iniziare a cancellare.

Digitate 4 e inserite il numero dell'ultima linea da eliminare. Inserite 6 e la routine cancellerà le linee dalla 4 alla 6. Ora potete salvare il programma in BASIC col comando SAVE "CANCELLINO" e il programma in linguaggio macchina con SAVE "CANCELLINO" CODE mem, 20. Date pure NEW, ora,

e caricate il programma da modificare, assicurandovi che non contenga linee 1,2,3 (eventualmente cancellatele momentaneamente). Caricate la routine in BASIC con MERGE". Fatto? Ora date GOTO 3 e caricate il linguaggio macchina. Per attivare la routine date GOTO 1 (non RUN!) e agite come già detto sopra.

Dimenticavamo: sulla cassetta troverete la routine già pronta per essere utilizzata, senza bisogno di dover effettuare la cancellazione delle linee 4/6.

5B04	2A005B	LD HL, (5B00)
5B07	CD6E19	CALL 196E
5B0A	E5	PUSH HL
5B0B	2A025B	LD HL, (5B02)
5B0E	23	INC HL
5B0F	CD6E19	CALL 196E
5B12	D1	POP DE
5B13	CDE519	CALL 19E5
5B16	C9	RET

Figura 1 - il disassemblato della routine di cancellazione in l/m

Cancellino versione per ZX Spectrum 16K/48K

```
1 INPUT "Cancella dalla linea ";f
  "alla linea ";t: IF f<3 OR t>999
  9 OR f>t THEN GO TO 1
2 POKE 23296,f-256*INT (t/256):
  POKE 23297,INT (f/256): POKE 23
  298,t-256*INT (t/256): POKE 2329
  9,INT (t/256): RANDOMIZE USR mem
  : STOP
```

```
3 LET mem=23300: LOAD "CANCELLINO"
  CODE mem,20: GO TO 1
4 LET mem=23300: LET cs=0:
  RESTORE 6: FOR i=0 TO 18:
  READ v: POKE (mem+i),v: LET cs=
  cs+v: NEXT i: READ csv: IF csv
  <>cs THEN PRINT FLASH 1;"Err
  ore nel listato": STOP
5 PRINT "Il CANCELLINO e' pronto!"
  : STOP
6 DATA 42,0,91,205,110,25,229,42,2
  ,91,35,205,110,25,209,205,229,25
  ,201,2081
```


Colspot

di E. E. Minetti.

Finestre a volontà in questo primo programma pubblicato da SUPERSINC per il QL.

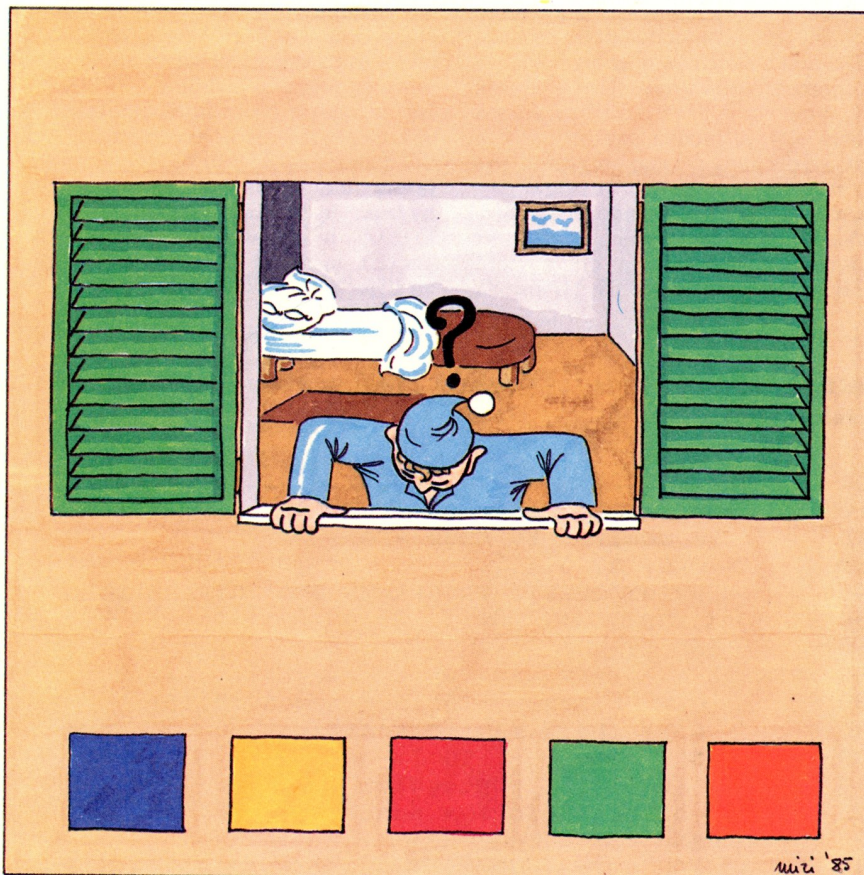
È ormai una consuetudine che i primi programmi pubblicati per computer da poco usciti sul mercato riguardino la grafica. Questo sia perché le prestazioni grafiche interessano la gran parte degli utenti sia perché le innovazioni tecnologiche coinvolgono quasi invariabilmente la risoluzione dello schermo e la velocità di esecuzione delle istruzioni. Il QL, ultima produzione della Sinclair, risponde a queste aspettative, alle quali aggiunge un linguaggio BASIC evoluto, il Super-Basic, in grado di rendere semplici delle sequenze di istruzioni, altrimenti complesse o noiose, il tutto a vantaggio della leggibilità del listato. La possibilità, inoltre, di incolonnare a piacimento le linee di programma (indenting), consente di evidenziarne la struttura logica (blocchi FOR...END FOR, IF...ELSE...END IF, ecc.). Un'altra novità di rilievo risiede nella possibilità di definire delle "procedure", sottoprogrammi dotati di variabili locali e richiamabili (dall'interno o dall'esterno del programma) semplicemente col nome a loro assegnato, eliminando completamente il ricorso alle classiche istruzioni GOTO e GOSUB (peraltro ugualmente utilizzabili).

Chi fosse completamente all'oscuro di questo nuovo linguaggio non si stupisca della mancanza delle istruzioni LET (inutile), THEN (sostituita da:) e NEXT (sostituita da END FOR).

Il programma qui proposto è un gioco basato sul riconoscimento dei colori e sulla rapidità dei riflessi, consistente nell'individuare in quale delle cinque finestre più piccole si trova il colore uguale alla finestra più grande.

Sarà necessario pertanto ricorrere ad una TV a colori, sia per la difficoltà di distinguere certe tonalità di grigio sia per godersi appieno le svariate tonalità di colore ottenibili sul QL.

Il gioco presenta sei livelli di difficoltà, a seconda del tempo concesso al giocatore per dare la sua risposta; questo è stato ottenuto grazie ad una opzione del comando INKEY\$, che permette di de-



finire il tempo di attesa del computer. Come si potrà notare, delle cinque procedure presenti nel programma l'ultima (forma__il__file) non viene chiamata dal programma e serve solamente per aprire il file "Classifica" sul dischetto del Microdrive 1. Essa va richiamata (battendo direttamente sulla tastiera "forma__il__file" seguito da ENTER solo prima di giocare per la prima volta, o qualora si volesse cancellare un'intera classifica e formarne una nuova, perdendo cioè nomi e punteggi dei precedenti giocatori).

Il gioco inizia alla linea 400: vengono aperte sei finestre, quanti sono i riquadri, più altre due finestre di cui una per il punteggio una per il numero delle mani giocate. Quindi inizia il loop in cui viene scelto in modo casuale il colore

per il riquadro più grande, viene assegnato il colore agli altri riquadri e vengono conteggiate le risposte esatte. Alla fine di questo loop, che si ripete 100 volte, compare la scritta "GAME OVER", il cui movimento è ottenuto alla linea 800 grazie alle istruzioni PAN e SCROLL, che muovono, rispettivamente in orizzontale ed in verticale, l'intero contenuto della finestra specificata.

Se il punteggio ottenuto è sufficiente per entrare in classifica, alla linea 820 viene richiamata la procedura "entri__in__classifica", dove viene messo in funzione il Microdrive 1 per cancellare la precedente classifica, già caricata e posta in array nella prima parte del programma, e ricrearne una nuova, inserendovi (linea 1140) la nuova sequenza

di punteggi e giocatori. Si torna quindi alla linea 830, dove viene chiamata la procedura "telescrivente". Anche qui si ricorre alla istruzione PAN, ma con una particolarità; infatti, non viene spostata verso sinistra tutta la finestra, che in questo caso è la finestra dei comandi alla base dello schermo (corrispondente al numero 0), bensì viene spostata solo la linea del cursore. Nell'istruzione PAN sono infatti specificati tre parametri: il primo corrispondente alla finestra entro la quale deve avvenire lo spostamento, preceduto dal simbolo "#" (cancellotto), il secondo indicante l'entità dello spostamento in pixel (valori positivi per spostamenti a destra e negativi per spostamenti a sinistra) e il terzo indicante quale porzione della finestra deve muoversi (se il valore è 0, o non è specificato, si muove tutto il contenuto della finestra; se è uguale a 3, si muove la linea del cursore; se è uguale a 4, si muove solo la parte a destra della linea del cursore, compreso il cursore). Un avvertimento: giocando ai livelli di difficoltà più alti, il tempo concesso per rispondere è veramente breve; sarà dunque opportuno prestare attenzione

ai suoni (differenziati per comparsa dei colori, risposta mancata o errata, risposta esatta), per non perdere il ritmo.

Descrizione del programma

Dalla linea 100 alla 210 viene aperto il file "Classifica", e vengono caricati e posti in due array rispettivamente, i punteggi ed i nomi dei giocatori. Come si può notare, entrambe le sequenze vengono caricate da dischetto con lo stesso nome (item\$).

Il file viene quindi richiuso (linea 210) e non verrà riaperto se non per un aggiornamento, nel caso che il giocatore abbia ottenuto un punteggio superiore a quello più basso presente nella classifica, quello cioè corrispondente al quindicesimo classificato.

Nella seconda parte viene allestito il menu iniziale e su richiesta viene chiamata la prima procedura: ISTRUZIONE. In questa procedura, non essendo specificato il tempo del comando PAUSE, (alla linea 1520), il programma si ferma fino a che venga premuto un qualsiasi tasto. Alla linea 340 viene dato il valore alla variabile "livello", che andrà a costituire il ritardo dell'istruzione INKEY\$ alla linea 670.

Nella terza parte (linea 400 - 760), vengono definite le finestre (OPEN...). Il primo numero dopo il simbolo "cancellotto" associa la finestra a un canale, e può variare da 0 a 15.

È bene però evitare i canali 0,1,2, perché già associati al normale output dello schermo; più precisamente, il canale 2 corrisponde alla parte superiore sinistra dello schermo, il canale 1 alla parte destra e il canale 0 alla parte inferiore, cioè la finestra dei comandi (utilizzando il QL con un televisore i canali 1 e 2 sono sovrapposti). In questo programma le finestre sono associate ai canali 5,6.. fino al 12. La linea 700 è responsabile del temporaneo cambiamento di colore del bordo del riquadro prescelto, che segnala l'esattezza della risposta.

Dalla linea 850 alla 890 vengono date le istruzioni per la scritta GAME OVER, e per il suo movimento, di cui si è già parlato; viene posta la condizione per l'ingresso in classifica e viene chiamata la procedura "telescrivente". Nella procedura "entri_in_classifica" troviamo l'istruzione OPEN_NEW, necessaria in questo caso per ricostituire ex novo il file classifica, che è appena stato cancellato dal comando DELETE.

Colspot versione QL

```

100 REMark -----
110   monitor_tv mode=PEEK(163890)
120   IF monItor_tv mode = 0
130     CLS: AT 5,5
140     PRINT"QUESTO PROGRAMMA PUO' GIRARE SOLO IN 'MODE TV'"
150     STOP
160   END IF
170 REMark -----
180   DIM hiscore (15):DIM nome$(15,15)
190   OPEN IN#13,mdv1 classifica
200   FOR gioc = 1 TO 15
210     INPUT#13, punteggio$
220     hiscore(gioc) = punteggio$
230   END FOR gioc
240   FOR vv = 1 TO 15
250     INPUT#13, punteggio$
260     nome$(vv) = punteggio$
270   END FOR vv
280   CLOSE#13
290 REMark -----
300   CLS: CLS#0: INK#0,7: CSIZE#0,0,0
310   AT#0,3,50:PRINT#0, "by E.E.Minetti 1985"
320   CSIZE 2,1: INK 176:PRINT: PRINT"           ColSpot "
330   CSIZE 2,0:INK 4:AT 8,2:PRINT "Per le istruzioni premi <I>."
340   AT 10,2:PRINT "Scegli il livello di difficolta"
350   AT 12,2 : PRINT "da uno a sei ....."!
360   car$ = INKEY$(-1): PRINT car$
370   IF car$ <>"i" AND (CODE(car$)<49 OR CODE(car$) >54 ): GO TO 300
380   IF car$ = "i" : istruzioni : GO TO 300
390   END IF
400   l = CODE(car$)-48
410   livello = 100/l
420   min score% = hiscore(gioc)+1/l
430   AT T4,2: PRINT "parte dopo il terzo segnale sonoro."

```

```

440 PAUSE 100:BEEP 5000,15:PAUSE 50:BEEP 5000,15:PAUSE 50:BEEP 5000,15
450 PAPER 0 :CLS:CLS#0
460 REMark -----
470 REMark ----- ColSpot E.E.MINETTI -----
480 REMark -----
490 OPEN#5,SCR 260X130A130,20:BORDER#5,4,176
500 OPEN#6,scr_80x40a45x160:BORDER#6,2,176
510 OPEN#7,scr_80x40a128x160:BORDER#7,2,176
520 OPEN#8,scr_80x40a215x160:BORDER#8,2,176
530 OPEN#9,scr_80x40a300x160:BORDER#9,2,176
540 OPEN#10,scr_80x40a385x160:BORDER#10,2,176
550 OPEN#11,scr_43x12a48x20:BORDER#11,1,176
560 OPEN#12,scr_43x12a428x20:BORDER#12,1,176
570 CSIZE#11,2,0:INK#11,4:CSIZE#12,2,0:INK#12,2
580 CSIZE#0,3,1:INK#0,4:PRINT#0," 1 2 3 4 5"
590 score = 0:PRINT#11,"000"
600 FOR mano = 1 TO 100
610 PRINT#12,"0"; mano
620 num = RND (29 TO 255)
630 PAPER#5,num
640 caso = RND (1 TO 5)
650 SElect caso
660 ON caso = 1:PAPER#6,num:PAPER#7,num-6:PAPER#8,num-12:PAPER#9,num-18:PAPER
#10,num-24
670 ON caso = 2:PAPER#7,num:PAPER#8,num-6:PAPER#9,num-12:PAPER#10,num-18:PAPE
R#6,num-24
680 ON caso = 3:PAPER#8,num:PAPER#9,num-6:PAPER#10,num-12:PAPER#6,num-18:PAPE
R#7,num-24
690 ON caso = 4:PAPER#9,num:PAPER#10,num-6:PAPER#6,num-12:PAPER#7,num-18:PAPE
R#8,num-24
700 ON caso = 5:PAPER#10,num:PAPER#6,num-6:PAPER#7,num-12:PAPER#8,num-18:PAPE
R#9,num-24
710 END SElect
720 FOR fin = 5 TO 10
730 CLS#fin: BEEP 2000, fin*2
740 END FOR fin
750 in$ = INKEY$(livello): risp = CODE(in$)
760 IF risp-48 = caso
770 BORDER#caso+5,2,4:BEEP 6000,7,5,5,5,14:PAUSE 12:BORDER#caso+5,2,176
780 score = score +1
790 PRINT#11 , "0"; score
800 ELSE
810 BEEP 4000,16,5,5,5,5
820 END IF
830 END FOR mano
840 PRINT#12,100
850 REMark -----
860 CSIZE#5 ,3,1: INK#5,176: AT#5,3,3: PRINT#5,"GAME OVER"
870 FOR p =20 TO 1 STEP -2:PAN#5,p:SCROLL#5,p:PAN#5,-p:SCROLL#5,-p:NEXT p
880 INK#0,7:INK 7:CSIZE 0,0
890 score = score * 1 : IF score > hiscore(gioc) : entri_in_classifica
900 telescrivente
910 REMark -----
920 DEFine PROCedure telescrivente
930 CSIZE#0,2,0
940 frase$=" ColSpot by E.E.Minetti PER GIOCARE PREMI 'P' PER VEDERE LA
CLASSIFICA PREMI 'C' "
950 REPEAT loop
960 FOR cell = 1 TO 90
970 con$ = frase$(cell)
980 AT#0,3,35: PRINT#0, con$;
990 PAN#0,-12,3
1000 IF con$<>" " : BEEP 400,5,5,5,1
1010 stop$ = INKEY$(0)
1020 IF stop$ = "p" : GO TO 300
1030 IF stop$ = "c" : presenta_la_classifica
1040 END FOR cell

```

```

1050 END REPEAT loop
1060 END DEFINE telescrivente
1070 REMARK -----
1080 DEFINE PROCEDURE entri_in_classifica
1090 DELETE mdv1_classifica : OPEN NEW#13,mdv1_classifica
1100 BEEP 4000,5: CSIZE#0,2,0: CLS#0: BEEP 4000,7
1110 AT#0,0,1: PRINT#0,"Con"!score!"punti,entri in classifica !"
1120 BEEP 7000,5: AT#0,1,1: INPUT#0,"Scrivi il tuo nome :",nuovo_nome$
1130 FOR c = 14 TO 0 STEP -1
1140     IF score > hiscore(c)
1150         hiscore(c+1)= hiscore(c)
1160         nome$(c+1) = nome$(c)
1170     ELSE EXIT c
1180     END IF
1190 END FOR c
1200 hiscore(c+1) = score : nome$(c+1) = nuovo_nome$
1210 FOR v = 1 TO 15: PRINT#13,hiscore(v): NEXT v
1220 FOR v = 1 TO 15: PRINT#13,nome$(v): NEXT v
1230 CLOSE #13
1240 END DEFINE entri_in_classifica
1250 REMARK -----
1260 DEFINE PROCEDURE presenta_la_classifica
1270 PAPER#5,33:CLS#5
1280 OVER 1 : CSIZE 2,0: INK 7
1290 FOR g = 1 TO 15
1300     AT g,5: PRINT ;nome$(g)
1310     AT g,16:PRINT g;" "
1320     AT g,20:PRINT"punti...."!hiscore(g)
1330 END FOR g
1340 OVER 0: CSIZE 0,0
1350 END DEFINE presenta_la_classifica
1360 REMARK -----
1370 DEFINE PROCEDURE istruzioni
1380 CLS : CSIZE 0,0: INK 7
1390 AT 0,27: PRINT "Premere il numero corrispondente al riquadro"
1400 AT 1,27: PRINT "che mostra il colore uguale al riquadro piB
1410 AT 2,27: PRINT "grande. Tasti 1 2 3 4 5 .La giusta risposta"
1420 AT 3,27: PRINT " segnalata da un 'beep' particolare e dal"
1430 AT 4,27: PRINT "bordo del riquadro che s'illumina di verde."
1440 AT 6,1: PRINT "Il punteggio finale dato dal numero"
1450 AT 7,1: PRINT "di risposte giuste (che appare in verde"
1460 AT 8,1: PRINT "in alto a sinistra durante il gioco),"
1470 AT 9,1: PRINT "moltiplicato per il grado di difficult
1480 AT 12,32: PRINT "Se appaiono due colori uguali in due"
1490 AT 13,32: PRINT "quadri vicini,scegli quello a sinistra"
1500 AT 16,1: PRINT "Non interrompere il gioco (CTRL\BARRA)"
1510 AT 17,1: PRINT "mentre il microdrive in funzione."
1520 AT 19,27: PRINT "Per tornare al gioco premi qualsiasi tasto"
1530 PAUSE
1540 END DEFINE istruzioni
1550 REMARK -----
1560 DEFINE PROCEDURE forma_il_file
1570 DELETE mdv1_classifica
1580 OPEN NEW #13,mdv1_classifica
1590 punti = 0
1600 FOR volte = 1 TO 15
1610     PRINT#13, punti
1620 END FOR volte
1630 stringa$ = ". . . ."
1640 FOR volte = 1 TO 15
1650     PRINT #13, stringa$
1660 END FOR volte
1670 CLOSE#13
1680 COPY mdv1_classifica TO scr_
1690 END DEFINE
1700 REMARK -----

```

Conversione basi numeriche

di **Guido Bertolotti**

Come si scrive 399 in base 32?

Questo programma ha lo scopo di convertire la rappresentazione di un numero in rappresentazioni in altre basi numeriche a scelta (da base due a base trentasei).

Il programma è stato scritto per ZX81, ma con alcune piccole modifiche gira anche sullo Spectrum.

Il problema della conversione numerica è per ogni utilizzatore di computer un aspetto importante dell'arte della programmazione, specialmente se programma in linguaggio macchina o vuole definire dei caratteri grafici (UDG). Prima di passare alla descrizione del programma è quindi utile richiamare qualche nozione teorica riguardante questo argomento.

Se noi abbiamo un insieme formato, per esempio, da centodiciassette oggetti, per indicarne il numero scriviamo così: 117. Con questo intendiamo che l'insieme è formato da un centinaio + una decina + sette unità di oggetti: la cifra più a destra ha il "peso" più basso (cifra delle unità) e tutte le altre hanno un "peso" dieci volte superiore a quello delle cifre immediatamente a destra. Si dice quindi che noi contiamo in base dieci.

Nulla però ci vieta di contare in altre basi, per esempio in base cinque. Il numero centodiciassette ha questa rappresentazione in base cinque: 432. Con questo intendiamo che l'insieme è formato da quattro venticinquine, tre cinque e due unità. Il numero degli oggetti è sempre lo stesso, ma è cambiato il modo di raggrupparli o, che è lo stesso, di rappresentarne il numero.

Chiarito questo, ci chiediamo ora: come si può "tradurre" la rappresentazione di un numero in una certa base nella sua rappresentazione in un'altra base? Il programma, nella sua brevità, permette facilmente di risalire all'algoritmo di conversione. Dopo essersi presentato il programma richiede le due basi e le memorizza nelle variabili B e C.

Richiede poi la rappresentazione da convertire e la memorizza in N\$.

Il seguente loop FOR A... (linea 310) trasferisce in N il valore numerico di



N\$. La variabile D, contenuta nel loop, ha il valore numerico della cifra N\$ (A). Il valore di D viene controllato alla linea 320 e, se è illecito (esempio: in base due sono lecite solo le cifre 0 e 1), si torna alla richiesta di N\$.

Supponiamo ora che il programma stia convertendo la rappresentazione 112 in base tre nella corrispondente rappresentazione in base quattro (vedi anche la fig. 1). Al primo passaggio del loop N avrà valore uno (vedi linea 330), cioè il numero delle "novine". Al secondo passaggio N viene moltiplicato per tre, e assume così il valore di tre (le terne contenute in una novina), a cui viene sommata la terna corrispondente alla seconda cifra della rappresentazione (N=quattro terne). Al terzo e ultimo

passaggio le quattro terne vengono moltiplicate per tre, ottenendo dodici unità, a cui si aggiungono le ultime due unità (cifra "2" di 112), ottenendo infine N=quattordici unità.

Terminato il loop si controlla che il valore di N non superi 4'294'967'295, è il più grande numero intero che può essere contenuto senza approssimazioni in una variabile. Se non esistesse questo controllo, si potrebbero ottenere, traducendo numeri troppo grandi, dei risultati approssimati.

Si passa poi alla seconda parte del programma, che inizia alla linea 390 e genera la rappresentazione in base C di N. Ritornando all'esempio scelto, alla linea 400 si divide quattordici per quattro, ottenendo tre, cioè il numero delle quar-

tine contenute in quattordici. Ma (vedi linea 401) tre per quattro fa dodici, e due unità restano fuori da questo raggruppamento, costituendo così la cifra delle unità della rappresentazione che vogliamo ottenere. Al secondo passaggio dividiamo le tre quartine in quattro, ottenendo zero esadecine e tre quartine, che restano fuori e costituiscono quindi la cifra delle quartine. Visto (linea 430) che non ci sono esadecine da raggruppare, viene stampato il risultato (32) e il programma passa alla richiesta di un nuovo numero (meglio: di una nuova rappresentazione).

Val forse la pena di ricordare che, se si vogliono rappresentare numeri in basi superiori a dieci, le dieci cifre che siamo abituati a usare non bastano più, e quindi si ricorre alle lettere dell'alfabeto (A vale dieci, B undici, ecc.), ma poiché a un certo punto anche le lettere finiscono, siamo costretti (a meno di voler usare altri simboli) a fermarci: è questo il motivo per cui il programma non può trattare numeri scritti in basi superiori a trentasei.

Oltre a rivelarsi utile per le usuali conversioni fra i sistemi binario, decimale ed esadecimale, questo programma ha una curiosa possibilità: predisponendolo a funzionare da base trentasei a base, diciamo, dieci, si possono codificare brevi parole (che, naturalmente, sono per il computer numeri in base trentasei) in incomprensibili gruppi di cifre decimali. Ovviamente, con la trasformazione inversa si ottengono nuovamente parole comprensibili.

Descrizione delle variabili

B	La base in cui è scritta la rappresentazione da tradurre.
C	La base in cui sarà scritta la rappresentazione tradotta.
N\$	Contiene le rappresentazioni: fino alla linea 390 la rappresentazione da tradurre, poi la rappresentazione tradotta. Il numero corrispondente alle rappresentazioni N\$.
N	Il numero corrispondente alle rappresentazioni N\$.
A	Variabile di controllo del loop, che trasferisce in N il valore di N\$: punta uno dopo l'altro i caratteri di N\$.
D	Il valore della cifra in esame: alla linea 315 viene ricavata dal carattere N\$(A) e alla 401 dal resto della divisione N/C.
E	Il risultato della divisione N/C (intero).

Descrizione del programma

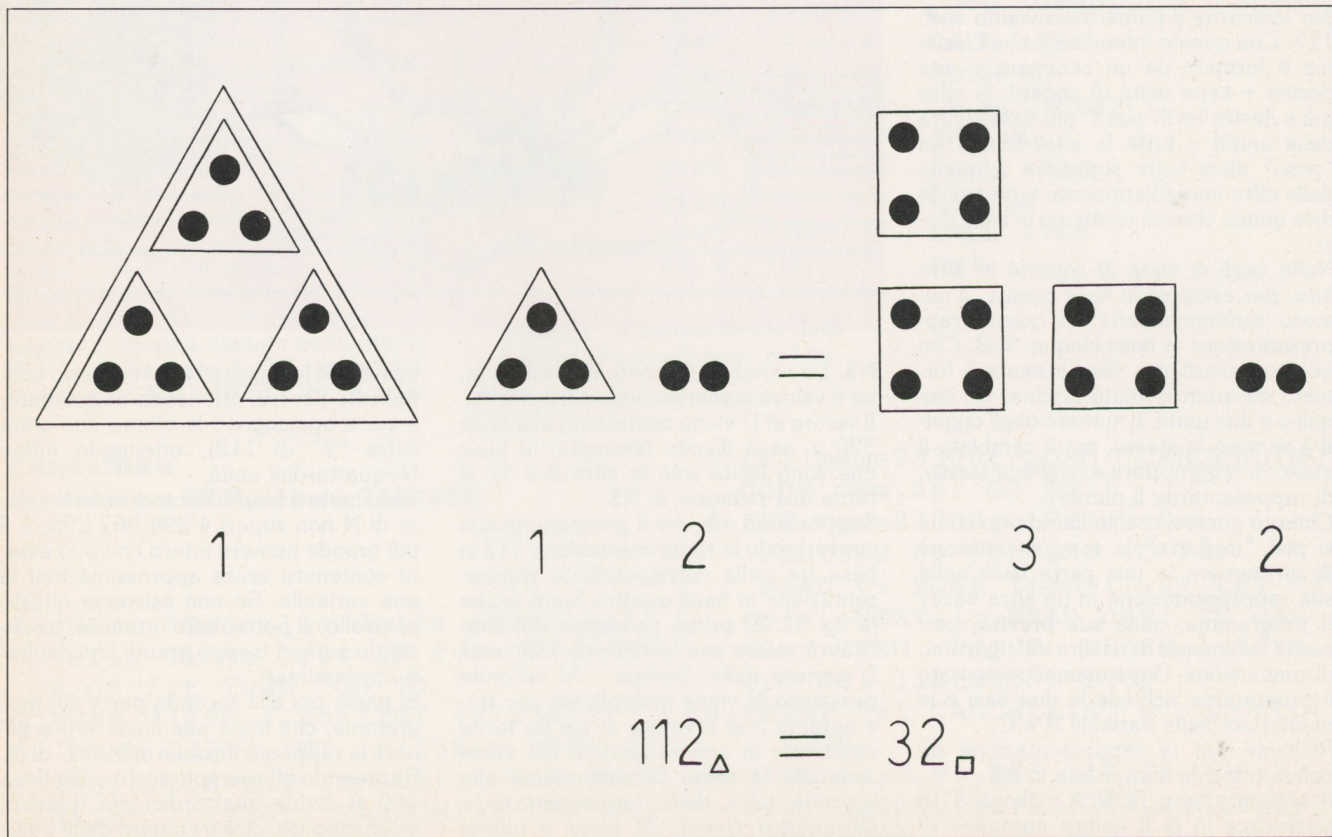
10-120	Commento.
130-270	Input dati.
300-340	Trasferimento in N del valore di N\$ (trasformazione da rappresentazione a numero).
350	Controllo overflow.
390-430	Trasferimento in N\$ del valore di N (trasformazione da numero a rappresentazione)
450-470	Output

Note importanti per gli utenti di ZX Spectrum

Per poter girare sullo Spectrum il programma necessita di alcune modifiche:
 1) eliminare tutte le linee contenenti le istruzioni SCROLL, FAST e SLOW (impossibili del resto da digitare, perché mancano sullo Spectrum);
 2) le linee del listato contrassegnate da REM \$\$ (\$\$=Solo Spectrum) vanno chiaramente scritte, eliminando il REM

\$\$;
 3) sostituire le ultime due linee con quest'unica:
 500 SAVE "convers." LINE 1

Noi possiamo rappresentare il numero di quattordici in anti-modi: per esempio in base tre o in base quattro. Quello che cambia è il modo di raggruppare gli elementi: il loro numero invece non cambia.



Conversione basi numeriche
versione per ZX81 16K.

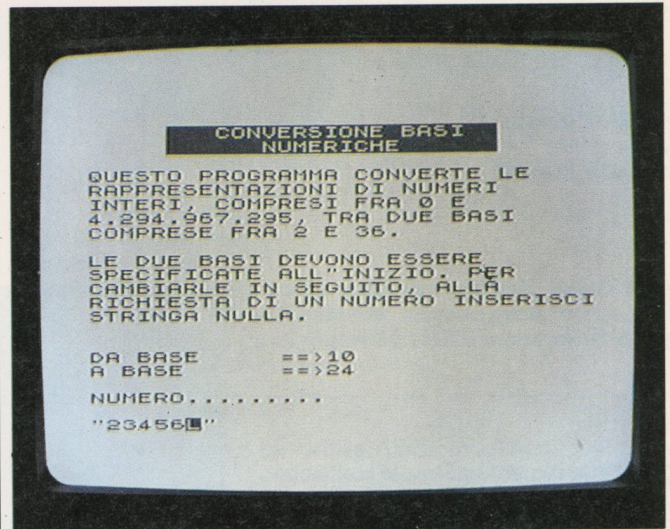
```

10 REM CONVERSIONE BASI
20 REM NUMERICHE
30 REM
40 REM DI GUIDO BERLOTTI
50 REM 4/1984
60 REM
70 SLOW
80 PRINT AT 5,5;" CONVERSIONE
E BASI";TAB 5;" NUMERICHE
90 PRINT
100 PRINT "QUESTO PROGRAMMA CON
VERTE LE RAPPRESENTAZIONI DI
NUMERI INTERI, COMPRESI FRA
0 E 4.294.967.295, TRA D
UE BASI COMPRESI FRA 2 E 36.
110 PRINT
120 PRINT "LE DUE BASI DEVONO E
SSERE SPECIFICATE ALL'INI
ZIO. PER CAMBIARLE IN SEGUIT
O, ALLA RICHIESTA DI UN NUM
ERO INSERISCI STRINGA NULLA."
130 PRINT AT 21,0;"DA BASE
==>";
140 INPUT B
150 IF B<2 OR B>36 OR B<>INT B
THEN GOTO 140
160 PRINT B
170 SCROLL
180 PRINT "A BASE ==>";
190 INPUT C
200 IF C<2 OR C>36 OR C<>INT C
OR C=B THEN GOTO 190
210 PRINT C
220 SCROLL
222 REM $$ PRINT
223 REM $$ PRINT
230 SCROLL
240 SLOW
250 PRINT AT 21,0;"NUMERO.....
..";
260 INPUT N$
270 IF N$="" THEN GOTO 130
280 FAST
290 PRINT AT 21,0;"
300 LET N=0
310 FOR A=1 TO LEN N$
311 REM $$ IF CODE N$(A)>=97 AN
D CODE N$(A)<=122 THEN LET N$(A)
=CHR$(CODE N$(A)-32): REM TRASF
ORMA LE MINUSCOLE IN MAIUSCOLE
315 LET D=CODE N$(A)-CODE "0"
316 REM $$ IF D>=10 AND D<=16 T
HEN GOTO 240: REM CONTROLLO SIMB
OLI ILLECITI
317 REM $$ IF D>9 THEN LET D=D-
7: REM SHIFT DI 7 CODICI IN MENO
DELLE LETTERE
320 IF D<0 OR D>=B THEN GOTO 24
0
330 LET N=N*B+D
340 NEXT A
350 IF N>4294967295 THEN GOTO 2
40
360 PRINT AT 21,0;N$;
370 IF LEN N$>=16 THEN SCROLL
380 PRINT "(BASE ";B;") ="
390 LET N$=""
400 LET E=INT (N/C)
401 LET D=N-E*C
402 REM $$ IF D>9 THEN LET D=D+
7: REM SHIFT DI 7 CODICI IN PIU"

```

LOAD

Conversione basi numeriche



```

DELE LETTERE
410 LET N$=CHR$(D+CODE "0")+N$
420 LET N=E
430 IF N<>0 THEN GOTO 400
440 SCROLL
450 PRINT N$;
460 IF LEN N$>=16 THEN SCROLL
470 PRINT "(BASE ";C;")"
480 GOTO 220
490 SAVE "CONVERSIONE"
500 RUN

```

La cassetta allegata a questo numero di Super Sinc contiene in omaggio un programma realizzato per lo Spectrum 16/48K:

"CODICE FISCALE"

di Gennaro Esposito,
del Gruppo Utilizzatore Computer Sinclair
Napoli
c/o Roberto Chimenti
via Luigi Rizzo, 18
80124 Napoli (NA)
Tel. 081/7623121

Supercomposer

di **J.W. Casson**
trad. e adatt.
di **Paolo Maffei**

Tutti come Beethoven con questo nuovo programma musicale per lo Spectrum 16K/48K

Sebbene soggetto alle note limitazioni sonore dello Spectrum, questo programma svolge in maniera efficace la funzione di scrivere musica su un pentagramma e di ascoltarne la melodia.

Come usare il programma

– **CHIAVE:** all'inizio il computer vi chiederà quali note desiderate vengano suonate, e se le volete diesis oppure bemolle.

Alla domanda "DIESIS?" risponderete quindi introducendo, una alla volta, le note che desiderate rendere diesis, oppure premendo solamente ENTER, se non ne desiderate alcuna.

Utilizzerete lo stesso procedimento alla domanda "BEMOLLE?".

Ricordo che rendere una nota diesis significa aumentarne l'altezza di un semitono; al contrario, rendere una nota bemolle significa diminuire la sua altezza di un semitono.

– **TEMPO:** vi verrà quindi richiesto il tempo di battuta. Dovrete allora introdurre, sotto forma di frazione, il tempo che desiderate: 3/4 per un valzer, 3/8 per una gavotta, 4/4 per la musica moderna, ecc.

– **SELEZIONE DEI CARATTERI:** si possono ora definire i caratteri musicali; le possibili scelte a disposizione sono:

PAUSA un periodo di silenzio
NOTA
DIESIS
BEMOLLE

Normalmente la scritta "nota" lampeggia nella griglia; per ottenere una pausa dovrete premere il tasto R fino a che la scritta corrispondente non lampeggerà; per tornare al carattere NOTA sarà sufficiente premere il tasto N.
Per ottenere una nota diesis o bemolle



Descrizione del programma

Linee

1
2-160

300-370
400-490
500-550
1000-1230
1500-1509

1510-1541
2010-2500
2500-2530

Commento

linguaggio macchina per lo scroll verso sinistra
inizializzazione delle variabili e determinazione dei caratteri maiuscoli
ciclo input dei caratteri musicali
ciclo input dei toni
disegno della nota e scroll
input delle note diesis, bemolle e del tempo di battuta
disegno della parte iniziale del pentagramma e della chiave di sol
disegno delle linee del pentagramma
viene compilata la melodia
suono della melodia

userete rispettivamente i tasti S o F e per cancellare questa scelta premerete il tasto opposto a quello selezionato (p.e. se lampeggia la scritta diesis, premerete il tasto F per tornare al carattere nota).

- DURATA DELLA NOTA: la durata della nota viene selezionata tramite i tasti 6 e 7, muovendo cioè il cursore lungo il menu fino a trovare il tipo di nota desiderato.

Effettuata la scelta premerete ENTER.

- SELEZIONE DEI TONI: a questo punto un altro cursore comparirà alla destra del pentagramma e potrà essere posizionato usando ancora i tasti-cursore 6 e 7 e quindi ENTER.

La nota verrà così disegnata nella giusta posizione nel rigo e il computer tornerà nella situazione di selezione dei caratteri musicali.

In questa situazione avete inoltre a disposizione le seguenti opzioni:

'P' - suona la melodia introdotta dall'inizio alla fine.

'K' + numero - lista il pentagramma dalla nota desiderata alla fine.

CAPS + '0' - cancella l'ultima nota introdotta.

La pressione del tasto 'D' durante la selezione dei toni vi permette di ritornare alla selezione dei caratteri musicali.

Come introdurre il programma

Per prima cosa bisogna introdurre la routine in linguaggio macchina, che per-

3000-3099
4000-4060
8500-9000
9500-9560

coordinate per disegnare le note, le pause, ecc.
disegno delle pause, scroll verso sinistra
coordinate per il LIST del pentagramma
subroutine per il disegno delle note e delle pause

Descrizione delle variabili

Tim	tempo di battuta
BA	durata delle note nella battuta
T(4,100)	contiene le informazioni della melodia come segue:
T(1,100)	tipo di nota o di pausa e
T(2,100)	altezza della nota, così come sono state introdotte
T(3,100)	nota o pausa effettiva e
T(4,100)	reale altezza della nota, così come sono state compilate
N(1,19)	DATA per convertire numeri in istruzioni BEEP
SS(100)	controllo dei caratteri - diesis, bemolle e sbarrette -
R	variabile binaria pausa
S	variabile binaria diesis
F	variabile binaria bemolle
C	variabile cursore
L	numero delle note introdotte
FN I	converte le INKEY\$ nella var. C
FN B	converte il tipo di nota in durata dei BEEP
FNP	converte T(2,L) in coordinate schermo

mette lo scroll del pentagramma verso sinistra. Questa routine è contenuta in una linea REM seguita da 32 caratteri.

Copiate quindi il programma n. 1 e date il RUN; per controllare che tutto funzioni perfettamente fate la seguente prova:

PRINT AT 9,10; "ABC": FOR I = 1 TO 9: RANDOMIZE USR sc: NEXT I.

Le lettere ABC dovrebbero comparire e muoversi verso sinistra.

Se la routine non dovesse funzionare correggete gli eventuali errori di copiatura, altrimenti eliminate, una alla volta, le linee 2, 9900, 9910, 9920. NON CANCELLATE LA LINEA 1.

Potete ora introdurre il pentagramma principale e dare libero sfogo al vostro talento musicale.

Supercomposer versione per ZX Spectrum 16K/48K

Listato 1 programma principale

```

1 REMH!H GO SUB VAL @>
!Hw GO SUB Z CLS <>.....
10 POKE 23658,8:: INK 0: PAPER 6:
    BORDER 6: BRIGHT 0: CLS : DIM S
    $(100): LET S=0: LET F=0
11 POKE 23676,255
13 DEF FN I(C,L,H)=C+(INKEY$="6")*(
    C<H)-(INKEY$="7")*(C>L)
25 DEF FN P(X)=50+2*X
30 LET SC=PEEK 23635+256*PEEK 23636
    +5
100 DIM N(19)
110 DIM T(4,100): LET L=1
130 GO SUB 1500
131 LET R=0
132 LET BA=0
133 DEF FN B(T)=2/2↑T
135 RESTORE
140 FOR N=1 TO 19: READ C
150 LET N(N)=C
160 NEXT N
170 DATA -5,-3,-1,0,2,4,5,7,9,11,12,
    14,16,17,19,21,23,24,26
180 LET C=1: LET OC=C: LET OPC=1:
    LET PC=OPC

```

```

190 GO SUB 1000
301 LET R=0: LET E=L
310 PRINT #0;AT 1,0;" 1 Semibreve
    2 Minima
    3 Semiminima
    4 Croma
    5 Semicroma"
320 PRINT #0;AT C,0;">": OVER 0
321 BRIGHT 1: PRINT AT 2,0; FLASH F;
    "BEMOLLE"; FLASH NOT R;AT 3,0;"N
    OTA "; FLASH S;AT 4,0;"DIESIS
    "; FLASH R;AT 3,7;" PAUSA ";
    FLASH 0;" :NOTA ";L;" ":
    BRIGHT 0
322 LET F=(NOT R) AND ((INKEY$="F"
    AND S=0) OR ((F AND INKEY$<>"
    S"))): LET S=NOT R AND (((
    INKEY$="S" AND F=0) OR (S AND
    INKEY$<>"F"))
330 IF INKEY$=CHR$ 13 AND BA+FN B(C)
    <=TIM THEN LET T(1,L)=C: BEEP .
    1,-30: LET BA=BA+FN B(C): LET S$
    (L)=CHR$ ((128*(BA=TIM)): LET B
    A=BA*(BA<TIM): GO TO 400+3600*R
331 IF INKEY$="K" THEN INPUT "LIST
    ";NL: LET L=NL: LET E=100:
    GO SUB 8500: GO TO 300
332 IF INKEY$=CHR$ 12 THEN LET L=L-
    1: LET BA=BA-FN B((T(1,L))-10*(T

```

```

(1,L)>10)): LET BA=BA+TIM*(BA<0)
: BEEP .1,-30: LET T(1,L)=0:
LET S$(L)="" : GO TO 300
333 IF INKEY$="P" THEN GO SUB 2000:
GO TO 300
334 LET R=((INKEY$="R") AND (R=0))
OR (R=1) AND (INKEY$<>"N")
340 LET C=FN I(C,1,5): IF OC<>C
THEN BEEP .05,-30: PRINT #0;
AT OC,0; OVER 1;">": LET OC=C
370 GO TO 320
401 OVER 1
402 LET S$(L)=CHR$(CODE S$(L)+50*S+
51*F)
410 PLOT 250,88-PC*2: DRAW -2,2:
DRAW 2,2
420 LET PC=FN I(PC,1,19)
425 IF INKEY$=CHR$ 13 THEN GO TO 47
0
430 IF OPC<>PC THEN BEEP .05,-30:
OVER 1: PLOT 250,88-OPC*2:
DRAW -2,2: DRAW 2,2: LET OPC=PC
: GO TO 410
435 IF INKEY$="D" THEN PLOT 250,88-
2*PC: LET BA=BA-FN B(T(1,L)):
LET BA=BA+TIM*(BA<0): DRAW -2,2
: DRAW 2,2: GO TO 300
460 GO TO 420
470 PLOT 250,88-PC*2: DRAW -2,2:
DRAW 2,2: LET T(2,L)=20-PC
480 OVER 0: GO SUB 1510: REM DRAW ST
AVE
490 RANDOMIZE USR SC
500 LET LI=L: GO SUB 3000
550 LET L=L+1: GO TO 300
1010 INPUT "DIESIS ?"; LINE I$
1015 GO SUB 9700
1020 IF I$="" THEN GO TO 1100
1030 LET S=(CODE I$)-63
1045 IF I$="G" THEN LET N(1)=N(1)+1
1060 FOR M=1 TO 19
1070 IF (M=S) OR (M=S+7) OR (M=S+14)
THEN LET N(M)=N(M)+1
1075 NEXT M
1090 GO TO 1010
1100 REM INPUT BEMOLLE
1110 INPUT "BEMOLLE ?"; LINE I$
1112 GO SUB 9700
1115 IF I$="" THEN GO TO 1200
1130 LET S=(CODE I$)-63
1150 IF I$="G" THEN LET N(1)=N(1)-1
1160 FOR M=1 TO 19
1170 IF (M=S) OR (M=S+7) OR (M=S+14)
THEN LET N(M)=N(M)-1
1180 NEXT M
1190 GO TO 1100
1210 INPUT "Tempo (sotto forma di fra
zione) ";tim
1220 LET S=0
1230 RETURN
1500 REM 1st STAVE
1501 GO SUB 1510: OVER 0: DRAW -8,0:
DRAW 0,-16
1502 RANDOMIZE USR SC: GO SUB 1510:
1503 PLOT 238,58: DRAW 4,0,PI:
DRAW 0,27: DRAW 2,4,PI/12:

```

```

DRAW 2,-4,-PI/10: DRAW -4,-7:
DRAW 0,-15,PI/1.2: DRAW 1,12,
PI: FOR r=0 TO 2: GO SUB 1510:
NEXT r
1509 GO SUB 1510: RETURN
1510 FOR N=0 TO 4
1520 PLOT 239,62+4*N
1530 DRAW 8,0
1540 NEXT n
1541 RETURN
2010 INPUT "Durata minima della nota
";T
2020 FOR N=1 TO L-1
2024 IF T(1,N)=0 THEN LET L=N:
GO TO 2500
2025 IF T(1,N)>10 THEN LET T(3,N)=10
0/2*((T(1,N)-10)): GO TO 2050
2030 LET T(3,N)=T*2/(2*(T(1,N))+T/20
*((S$(N-(N>1))<CHR$ 128))
2040 LET T(4,N)=N(T(2,N))+S$(N)="2"
OR S$(N)="SIN ")-(S$(N)="3"
OR S$(N)="COS ")
2050 NEXT N
2499 PRINT #0;"TUNE COMPILED PRESS AN
Y KEY": PAUSE 0
2500 FOR N=1 TO L
2510 IF T(1,N)<10 THEN BEEP T(3,N),T
(4,N): NEXT N
2511 PAUSE (T(3,N)+1)
2520 NEXT N
2530 RETURN
3001 LET S=0: LET F=0
3010 GO SUB 1510
3015 LET R=0: IF T(1,L)>10 THEN
LET R=1: LET T(1,L)=T(1,L)-10
3017 IF S$(L)="2" OR S$(L)="SIN "
THEN PRINT AT 15-(T(2,L)/4),28
;"#": LET S=1
3018 IF S$(L)="3" OR S$(L)="COS "
THEN PRINT AT 15-(T(2,L)/4),28
;"b": LET F=1
3020 GO SUB 9000+100*T(1,L)+50*R
3030 REM NOTE ABOVE OR BELOW STAVE
3039 IF R THEN GO TO 3080
3040 IF T(2,L)>15 THEN PLOT 235,82:
DRAW 8,0
3050 IF T(2,L)>17 THEN PLOT 235,86:
DRAW 8,0
3060 IF T(2,L)<5 THEN PLOT 235,58:
DRAW 8,0
3070 IF T(2,L)<3 THEN PLOT 235,54:
DRAW 8,0
3080 RANDOMIZE USR SC
3090 FOR J=5+(S$(L+1)="3" OR S$(L+1)=
"SIN " OR S$(L+1)="2" OR S$(L+1)
="COS ") TO T(1,L) STEP -1:
GO SUB 1510: RANDOMIZE USR SC:
NEXT J
3091 IF CODE S$(L)>=128 THEN PLOT 24
0,62: DRAW 0,16: GO SUB 1510:
RANDOMIZE USR SC
3092 IF R THEN LET T(1,L)=T(1,L)+10:
LET R=0
3099 RETURN
4038 LET T(1,L)=T(1,L)+10
4039 GO SUB 3000

```

```

4040 LET L=L+1
4060 GO TO 300
8021: IF T(2,L)>18 THEN PLOT 235,90:
      DRAW 8,0
8500 REM LISTN,E 0
8510 RANDOMIZE USR SC: GO SUB 1500:
      REM TREBLE CLEF
8520 FOR X=0 TO 10
8535 LET L=X+NL
8537 IF T(1,L)=0 OR L>E THEN
      RETURN
8540 GO SUB 3000
8576 NEXT X
8580 IF L<E THEN LET NL=NL+X:
      PRINT AT 0,0;"Scroll?": PAUSE 0
      : PRINT AT 0,0;;; OVER 0;"
      ": GO TO 8520
8590 PRINT BA: PAUSE 0: RETURN
9101 OVER 0
9105 LET Y=FN P(T(2,L))
9110 CIRCLE 239,Y,2
9140 RETURN
9160 PLOT 245,70: DRAW 0,-2: DRAW 5,0
      : DRAW 0,1: DRAW -5,0: RETURN
9210 GO SUB 9100
9220 PLOT 241,Y
9230 DRAW 0,11
9240 RETURN
9260 PLOT 245,71: DRAW 0,1: DRAW 5,0:
      DRAW 0,-1: DRAW -5,0: RETURN
9310 LET Y=FN P(T(2,L))
9320 FOR D=0 TO 2 STEP .5
9330 CIRCLE 239,Y,D
9340 NEXT D

```

Supercomposer versione per ZX Spectrum 16K/48K

Listato 2 generatore del linguaggio macchina

```

1 REM .....
....
2 LET sc=PEEK 23635+256*PEEK 23636
+5

```



```

9349 PLOT 241,Y: DRAW 0,11: RETURN
9360 PLOT 245,66: DRAW 0,4: DRAW 2,0:
      DRAW -1,3: RETURN :
9400 REM QUAVER
9410 LET Y=FN P(T(2,L))
9420 FOR D=0 TO 2 STEP .5
9430 CIRCLE 239,Y,D
9440 NEXT D
9449 PLOT 241,Y: DRAW 0,11: DRAW 3,-2
      : DRAW -3,1: RETURN
9460 PLOT 245,70: DRAW 4,1: DRAW -2,-
      6: RETURN
9510 GO SUB 9400
9520 DRAW 0,-3: DRAW 4,-1: DRAW -4,1
9530 RETURN
9560 PLOT 245,70: DRAW 4,1: DRAW 0,-2
      : DRAW 3,1: DRAW -3,1: DRAW -2,-
      6: RETURN
9700 IF I$="DO" THEN LET I$="C"
9710 IF I$="RE" THEN LET I$="D"
9720 IF I$="MI" THEN LET I$="E"
9730 IF I$="FA" THEN LET I$="F"
9740 IF I$="SOL" THEN LET I$="G"
9750 IF I$="LA" THEN LET I$="A"
9760 IF I$="SI" THEN LET I$="B"
9770 RETURN

```

```

9900 FOR n=0 TO 26
9910 READ byte: POKE sc+n,byte:
      NEXT n
9920 DATA 1,0,8,17,0,72,33,1,72,237,1
      76,6,64,62,0,17,32,0,33,31,72,11
      9,237,90,16,251,201

```

Dal "Dizionario della disinformatica", pagina INT(RDN*1000)+1:

ADATTATORE

connettore usato per collegare al computer apparecchi con esso incompatibili. L'incompatibilità permane, ma almeno il collegamento esiste.

ALTA RISOLUZIONE

tecnica di programmazione avente lo scopo di far tracciare dal computer linee diagonali estremamente antiestetiche.

AMPLIFICATORE SONORO

dispositivo interfacciabile al computer che permette ai vicini di casa di sentire i vostri bambini giocare a "3D Pacvaders".

ARCADE GAME

coloratissimo e rumoroso misuratore di riflessi.

ARRAY

tipico urlo di gioia del programmatore che constata il perfetto funzionamento del suo programma per la prima volta.

AUTOSTART

meccanismo che provoca l'apertura del fuoco alieno prima che abbiate avuto il tempo di toccare i tasti di controllo della vostra astronave.

AVVENTURA

- 1) gioco di fantasia che prevede l'utilizzo in maniera logica di oggetti estremamente illogici.
- 2) caricamento di un lunghissimo programma sullo ZX81.

Undici

di **Colin Gooch**
trad. e adatt.
di **Andrea Violini**

Un simpatico passatempo per tutti gli Spectrum.

Questo è uno dei più semplici giochi della grande famiglia dei solitari che, pur basando la sua soluzione su una semplice disposizione casuale di carte, saprà divertirti e... farvi arrabbiare!

Il computer visualizza 9 carte, che vanno eliminate a coppie (quando la somma di ogni coppia risulti 11) o a terne (nel caso di una "scala" con fante, donna e re).

Se non si può continuare, si può scambiare casualmente la carta al centro. Se ancora non si può continuare, bisogna arrendersi.

La prima cosa che bisogna pensare prima di programmare un gioco di carte è la presentazione di queste ultime. Un mazzo di carte alte 9 spazi carattere e larghe 5 dà risultati veramente realistici, ma restringe sfortunatamente il numero di carte che possono essere mostrate: un massimo di 2 file da 4 (5 omettendo gli spazi fra una carta e l'altra). Questo gioco necessita di visualizzare contemporaneamente 9 carte, e così si è deciso di restringere le dimensioni di ogni carta a un blocco di 7 per 5 spazi carattere. In questa maniera i semi non possono essere disposti come dovrebbero, ma il risultato è comunque più che accettabile. Le carte sono rappresentate in memoria da 5 caratteri contenuti nella stringa P\$, dove vengono mischiate e distribuite.

I 5 caratteri identificano nell'ordine il "nome" della carta, il suo valore (2 caratteri), il seme e il colore.

Svolgimento del programma

Il programma viene salvato con auto-start dalla linea 1760, a partire dalla quale vengono definiti gli UDG, per poi passare alle parti riguardanti il gioco vero e proprio.

Le prime operazioni cominciano alla linea 1020, dove vengono definite le coordinate x e y relative alla stampa delle carte. Il ciclo principale, che è abbastanza lungo, parte dalla linea 1040. Prima di tutto dà la possibilità di leggere



Descrizione delle variabili

A	conta le carte distribuite
CD	carte giocate
FA	flag temporaneo
LU zero	usate per risparmiare memoria,
LV uno	consentendo al programma
LW due	di girare anche su 16K.
N	usata nei cicli FOR...NEXT
T	valore totale delle carte da eliminare
X, Y	posizione sullo schermo delle carte
Z (X)	valori derivati da Z\$
A\$, B\$, D\$	formazione del mazzo
C\$	carta stampata
E\$, F\$	componenti di C\$
P\$	mazzo
R\$ (x)	carte distribuite
Z\$	la mossa del giocatore
T\$	la scritta "undici" ingrandita
U\$	istruzioni del gioco

le istruzioni situate dalla linea 1600 in poi, e successivamente definisce alcune variabili (linea 1710), per poi preparare e mischiare il mazzo (linee 1400-1490).

La linea 1060 stampa una mano, usando la routine di stampa delle carte, presente alla linea 1250. Viene così richiesta al giocatore la mossa da eseguire. Le linee dalla 1090 alla 1210 controllano se la combinazione è valida. Se tutto è regolare, la mossa viene

stampata e si ricomincia dall'inizio.

Se l'inserimento non è valido, viene stampato un messaggio di errore, e si ripete l'input. In caso di esaurimento del mazzo, o di resa del giocatore, il programma aggiunge la routine di fine partita (linea 1510).

Lo Spectrum tiene il conto delle carte distribuite mediante la stringa R\$. Questa scelta si è resa necessaria per poter riconoscere il valore 10, che è rappresentato da un UDG, non identificabile

dalla funzione SCREEN\$.

La routine di stampa divide ogni carta nelle sue varie componenti (attraverso lo slicing), utilizzandole per procedere alla visualizzazione, alle coordinate precedentemente stabilite dalle funzioni c(x) e d(x), definite all'inizio del programma.

Le figure (fante, donna e re) sono state definite dalle lettere J, Q e K.

Undici versione per ZX Spectrum 16K/48K

```

1020 LET LW=2: LET LV=1: LET LU=0:
      DEF FN C(X)=VAL "6"*(X=VAL "2"
      OR X=VAL "6")+VAL "12"*(X=VAL "
      9")+VAL "18"*(X=VAL "3" OR X=
      VAL "7")+VAL "24"*(X=VAL "4"
      OR X=VAL "8"): DEF FN D(X)=
      VAL "2"+VAL "6"*(X>VAL "4")+
      VAL "6"*(X>VAL "4" AND X<VAL "9"
      )
1030 RANDOMIZE : RANDOMIZE
1040 GO SUB 1600: BORDER 4: PAPER 4:
      INK LU: CLS : GO SUB 1710:
      GO SUB 1400: GO SUB 1500
1050 REM STAMPA LA MANO DI
      PARTENZA
1060 LET A=LV: FOR N=LV TO 44 STEP 5:
      : LET X=FN C(A)+LW: LET Y=FN D(A
      )+7: PRINT AT Y,X; PAPER 6;
      INK 0;" ";A;" ": LET A=A+LV:
      NEXT N
1070 LET A=LV: FOR N=LV TO 44 STEP 5:
      LET X=FN C(A): LET Y=FN D(A):
      LET C$=P$(N TO N+4): LET R$(A)=
      C$(LW TO 3): GO SUB 1260: LET A=
      A+LV: NEXT N: LET CD=9
1080 LET P$=P$(46 TO )
1090 PRINT AT 18,13; PAPER 5; INK LU;
      "CARTE";AT 19,13;" NEL ";AT 20,1
      3;"MAZZO";AT 21,13;"      ";AT 21
      ,15;52-CD: INPUT AT LU,LU;"INSER
      ISCI LA POSIZIONE DELLE""CARTE
      DA ESTRARRE (0 RESA) "; LINE Z$:
      IF Z$="" THEN GO SUB 1230
1100 IF LEN Z$>3 THEN GO SUB 1230:
      GO TO 1100
1110 DIM Z(LEN Z$): FOR N=LV TO LEN Z
      $: IF CODE Z$(N)<48 OR CODE Z$(N
      )>57 THEN GO SUB 1230: GO TO 11
      00
1120 LET Z(N)=VAL Z$(N): NEXT N: IF
      LEN Z$=LV AND Z$(LV)<>"9"
      AND Z$(LV)<>"0" THEN
      GO SUB 1230: GO TO 1100
1130 IF LEN Z$=LW THEN LET FA=LU:
      IF Z(LV)=Z(LW) THEN GO SUB 123
      0: GO TO 1100
1140 IF LEN Z$=3 THEN LET FA=LU:
      IF (Z(LV)=Z(LW))+ (Z(LV)=Z(3))+
      (Z(LW)=Z(3)) THEN GO SUB 1230:
      GO TO 1100
1150 IF VAL Z$=LU THEN LET RES=LV:
      GO TO 1510
1160 IF FA THEN GO SUB 1230: GO TO 1
      100
1170 IF LEN Z$=LV THEN LET FA=FA+LV
1180 FOR N=LV TO LEN Z$: LET T=T+
      VAL R$(Z(N)): NEXT N: IF T<>11
      AND T<>36 AND LEN Z$<>LV
      THEN GO SUB 1230: GO TO 1100
1190 IF T=11 AND LEN Z$<>LW THEN
      GO SUB 1230: LET T=LU: GO TO 11
      00
1200 IF T=36 AND LEN Z$<>3 THEN
      GO SUB 1230: LET T=LU: GO TO 11
      00
1210 FOR N=LV TO LEN Z$: LET C$=P$(LV
      TO 5): LET P$=P$(6 TO ): LET CD
      =CD+LV: IF CD=52 THEN GO TO 151
      0: GO TO 1220: LET X=FN C(Z(N)):
      LET Y=FN D(Z(N)): GO SUB 1260:
      LET R$(Z(N))=C$
1220 LET X=FN C(Z(N)): LET Y=FN D(Z(N
      )): GO SUB 1260: LET R$(Z(N))=C$
      (LW TO 3): NEXT N: LET T=LU:
      GO TO 1090
1230 INPUT AT LU,LU;"INSERIMENTO NON
      VALIDO""INSERISCI ANCORA ";
      LINE Z$: IF Z$="" THEN GO TO 1
      230
1240 RETURN
1250 REM STAMPA LA CARTA
1260 INK VAL C$(5): LET E$=C$(4):
      LET F$=C$(LV)
1270 PAPER 7: FOR M=Y TO 8+12*(Y>10)+
      6*(Y=8): PRINT AT M,X+LV; INK LU
      ;"      ": NEXT M: PRINT AT Y,X+L
      V;F$;AT Y+6,X+5;F$
1280 PRINT AT Y+LV,X+LV;E$;AT Y+5,X+5
      ;E$
1290 IF (F$="A")+(F$="3")+(F$="5")+(F
      $="9") THEN PRINT AT Y+3,X+3;E$
1300 IF (F$="2")+(F$="3") THEN
      PRINT AT Y+LW,X+3;E$;AT Y+4,X+3
      ;E$
1310 IF (F$="4")+(F$="5")+(F$="8")+(F
      $="9")+(F$="A") THEN PRINT AT Y
      +LW,X+LW;E$;AT Y+LW,X+4;E$;AT Y+
      4,X+LW;E$;AT Y+4,X+4;E$
1320 IF (F$="6")+(F$="7") THEN
      PRINT AT Y+LV,X+LW;E$;AT Y+LV,X
      +4;E$;AT Y+3,X+LW;E$;AT Y+3,X+4;
      E$;AT Y+5,X+LW;E$;AT Y+5,X+4;E$

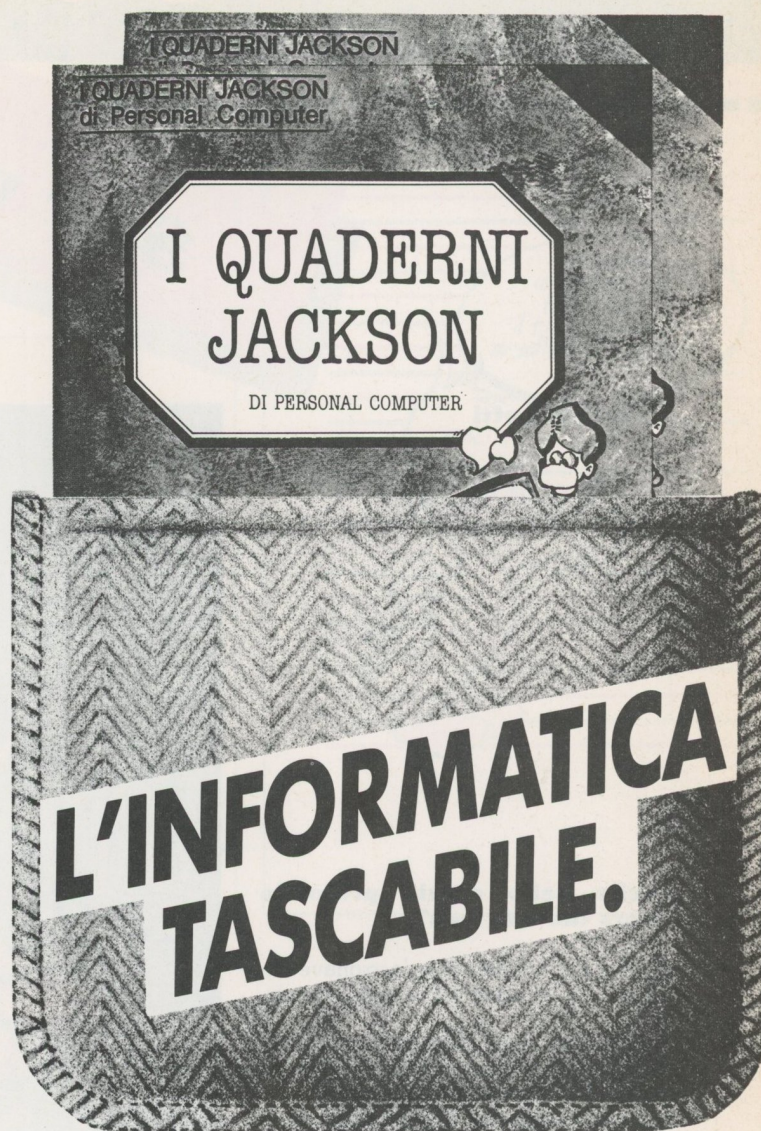
```



```

1640 CLS : PRINT AT LV,LV; INK LU;"UN
DICI - IL SOLITARIO: REGOLE":
PAUSE 100
1650 DIM U$(7,30): LET U$(LV)="SARANN
O DISTRIBUITE 9 CARTE:": LET U$(
LW)="DEVI ESTRARNE 2 ALLA VOLTA"
: LET U$(3)="(O UNA SERIE DI J-Q
-K)": LET U$(4)="TOTALIZZANDO OG
NI VOLTA 11": LET U$(5)="(ASSO=1
)": LET U=5
1660 GO SUB 1690: GO SUB 1500: LET U$
(LV)="SE NON PUOI CONTINUARE DEV
I": LET U$(LW)="SCAMBIARE (UNA V
OLTA SOLA) LA": LET U$(3)="CARTA
NUMERO 9 (POSTO CHE NON": LET U
$(4)="SIA UNA FIGURA). SE LA SOR
TE": LET U$(5)="TI E' ANCORA NEM
ICA, HAI PERSO": LET U$(6)="E DE
VI ARRENDERTI...": LET U=6
1665 GO SUB 1690: GO SUB 1500: LET U$
(LV)="PER SCEGLIERE LE CARTE DA"
: LET U$(LW)="TOGLIERE DEVI DIGI
TARNE I": LET U$(3)="NUMERI DI S
EGUITO. ESEMPIO:": LET U$(4)="56
= TOLGO LE CARTE NUMERO 5":
LET U$(5)="E 6. PER TOGLIERE UN
A FAMIGLIA": LET U$(6)="DI J-Q-K
SERVONO 3 NUMERI": LET U$(7)="(
ES.: 714 = CARTE NUM. 7,1,4).":
LET U=7
1670 GO SUB 1690: GO SUB 1500
1680 RETURN
1690 FOR N=LW TO 17: PRINT AT N,LV;
PAPER 5;"
": NEXT N
1700 LET B=LW: FOR N=4 TO 3+(U*LW)
STEP LW: FOR M=LW TO 30:
PRINT AT N,M; PAPER 7; INK LV;"
*": PRINT AT N,M; PAPER 5;U$(B,M
): BEEP .01,LU: NEXT M: LET B=B+
LV: NEXT N
1710 REM INIZIALIZZAZIONE
1720 LET A=LW: LET CD=LU: LET T=LU:
LET FA=LU: LET RES=LU
1730 DIM R$(9,5)
1740 LET B$="BDCE"
1750 RETURN
1760 REM GRAFICI (UDG)
1770 BORDER 2: CLS : PRINT AT 10,5;"F
ERMA IL REGISTRATORE": LET LU=0:
LET LV=1: LET LW=2
1780 RESTORE 1800: FOR N=LU TO 4:
FOR M=LU TO 7: BORDER M: BEEP .
01,N
1790 READ A: POKE USR CHR$(97+N)+M,A
: NEXT M: NEXT N: RUN
1800 DATA LU,94,82,82,82,82,94,LU
1810 DATA LU,16,56,124,254,124,56,16
1820 DATA LU,108,254,254,124,124,56,1
6
1830 DATA 16,56,124,124,254,214,84,16
1840 DATA LU,56,56,16,214,254,214,16
1850 RUN
9990 SAVE "UNDICI" LINE 1760

```



Quaderni Jackson:
l'informatica a tutti i livelli, in
una collana aperta, pratica,
essenziale, aggiornata.

Tutto quello che è
importante sapere sui
computer, la programmazione,
i linguaggi, il software, le
applicazioni e i nuovi sviluppi
dell'informatica.

Ogni mese, 2 volumi.

Volumi già pubblicati:

Gianni Giaccaglioni
"Vivere col Personal Computer"
Paolo Bozzola
"Dentro e fuori la scatola"

Enrico Odetti
"Ed è subito BASIC Vol. I"
"Ed è subito BASIC Vol. II"
Paolo Capobussi
e Marco Giacobazzi
"A ciascuno il suo Personal"
Fulvio Francesconi
e Fernando Paterlini
"To do or not to do"
Gianni Giaccaglioni
"Strutturare il software"
Enrico Odetti
"Dizionario informaticese"

In edicola,
a sole lire 6.000.



**GRUPPO
EDITORIAL
JACKSON**

SAN FRANCISCO - LONDRA - MILANO

Star war

di Luciano Lotti

Anche nello spazio profondo "mors tua vita mea".

Siete al comando di un'astronave dotata di un laser ad energia variabile.

Il vostro schermo segnala la posizione di oggetti nello spazio senza fare distinzione tra astronavi e meteoriti.

La vostra missione è di distruggere la flotta nemica, composta da dieci incrociatori e una nave ammiraglia.

Per annientare un incrociatore occorre un'energia di fuoco di 500 unità, mentre per l'ammiraglia ne sono necessarie 1000.

Per le meteoriti è sufficiente una qualunque intensità superiore a 0.

Se si attacca un'astronave avversaria con energia insufficiente, essa risponderà al fuoco.

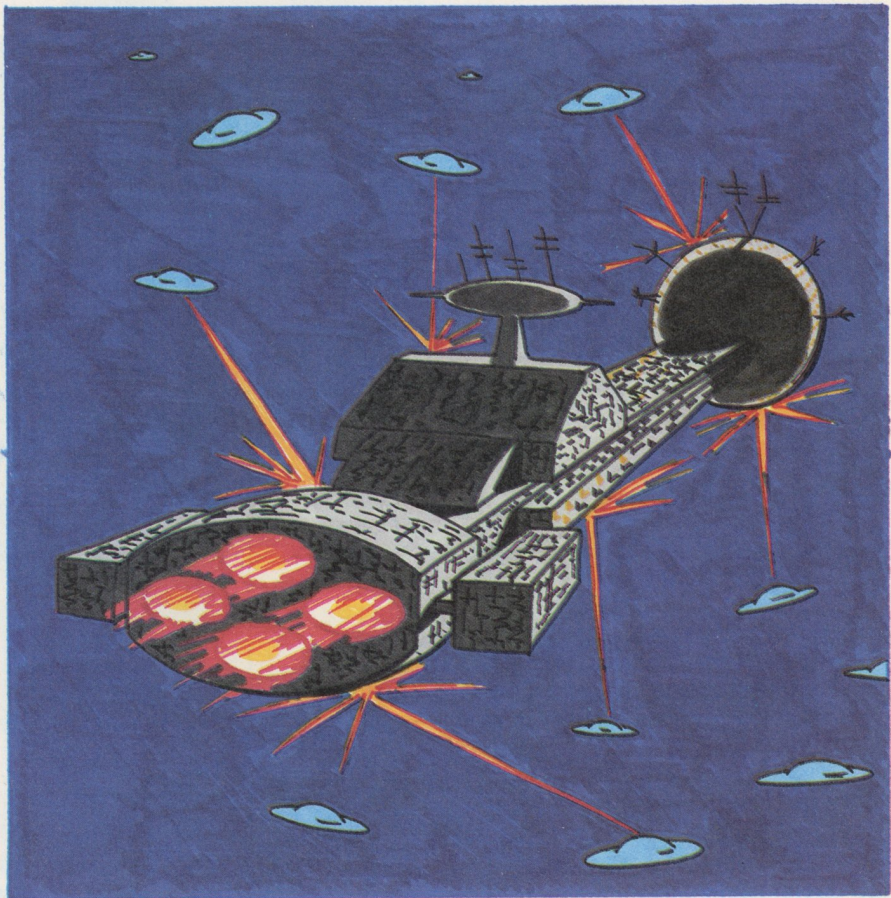
I colpi degli incrociatori, se vanno a segno, assorbono un po' d'energia; sono invece letali quelli dell'ammiraglia.

Dopo aver risposto al fuoco le astronavi si spostano.

Avete anche a disposizione un radar. Quando l'addetto alla sala radar si metterà in comunicazione con voi risponderetegli "roger" ed otterrete il numero di astronavi e di asteroidi presenti nella zona quadrata che cambierà colore sullo schermo.

Per pilotare il radar si devono indicare le coordinate della casella che si vuole al centro del quadrato analizzato.

Tenete sempre un occhio sull'energia disponibile perché, se termina, la partita subisce la stessa sorte.



Descrizione del programma

Linee	Commento
860-900	inizializzazione variabili
910-980	riempimento matrice con i valori corrispondenti alle astronavi ed agli asteroidi
1005	controlla se è terminata l'energia
1015-1070	input relativo al radar, alle coordinate e all'energia di fuoco e loro controllo
1090	fuoco su casella vuota
1100	colpiti degli asteroidi
1110-1140	attaccato e distrutto incrociatore
1150	energia insufficiente alla distruzione
1300-1310	colpita l'ammiraglia, ma con energia insufficiente
1320-1330	ammiraglia distrutta
5005-5025	reazione di un incrociatore

5030-5040 l'incrociatore si sposta
5500-5520 reazione dell'ammiraglia
5525-5540 l'ammiraglia si sposta
6000-6300 disegna il quadro
6500-6999 radar
7000-7120 disegna l'addetto al radar
9000-9990 procedure di fine

Descrizione delle variabili

i\$ usataper l'input
i\$ usata per cancellare parti del video
a contiene il numero della riga (trovato casualmente) in cui inserire un elemento in m()
b come a per le colonne
en energia rimasta
ener energia di fuoco
fla segnala alla routine di fine il messaggio da segnalare (3=missione non completata)
g variabile di FOR-NEXT usata in vari cicli
h variabile di FOR-NEXT usata (con g) per la stampa dello schermo
inc numero di incrociatori nella zona analizzata dal radar
incr numero di incrociatori rimasti
m() matrice contenente le astronavi e i meteoriti da visualizzare sullo schermo
met numero di meteoriti nella zona analizzata dal radar
mo numero di mosse fatte
x1 riga a cui si spara
y1 colonna a cui si spara



Star war

versione per ZX Spectrum 16K/48K

```

860 LET i$=""
      "
865 LET incr=10
870 BORDER 1: PAPER 1: INK 6: CLS
890 DIM m(9,9)
900 LET en=8000
910 FOR g=1 TO 10
920 LET a=INT (RND*9+1): LET b=INT (
      RND*9+1): IF m(a,b)<>0 THEN
      GO TO 920
930 LET m(a,b)=1: NEXT g
940 FOR g=1 TO 10
950 LET a=INT (RND*9+1): LET b=INT (
      RND*9+1): IF m(a,b)<>0 THEN
      GO TO 950
960 LET m(a,b)=2: NEXT g
970 LET a=INT (RND*9+1): LET b=INT (
      RND*9+1): IF m(a,b)<>0 THEN
      GO TO 970
  
```

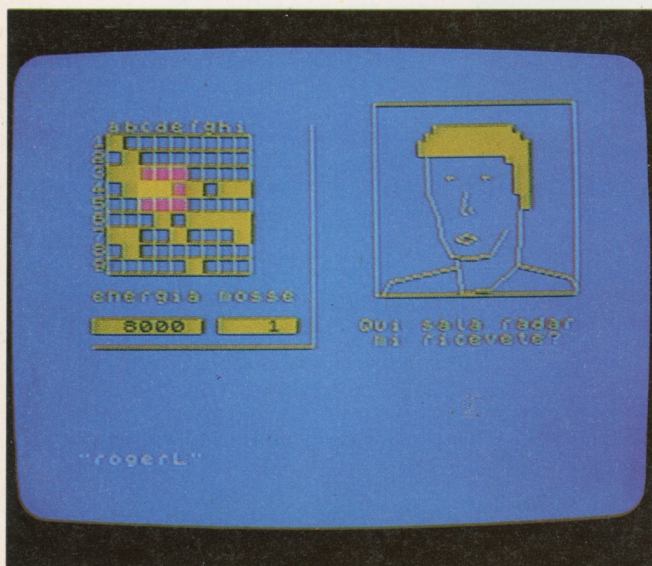
```

980 LET m(a,b)=3
990 LET mo=0
1000 LET mo=mo+1
1005 IF en<=0 THEN GO TO 9000
1010 GO SUB 6000
1015 INPUT " radar (s/n) ?"; LINE i$
1020 IF i$="s" THEN GO SUB 6500:
      REM radar
1030 INPUT "coordinate di fuoco ?";i$
1035 IF LEN i$<>2 THEN GO TO 1030
1040 IF CODE i$(1)<49 OR CODE i$(1)>5
      7 OR CODE i$(2)<97 OR CODE i$(2)
      >105 THEN GO TO 1030
1050 LET y1=VAL i$(1): LET x1=CODE i$
      (2)-96
1060 INPUT "energia ?";ener
1070 IF ener>en OR ener<=0 THEN
      GO TO 1060
  
```

```

1080 LET en=en-ener
1090 IF m(y1,x1)=0 THEN GO TO 1000
1100 IF m(y1,x1)=1 THEN PRINT AT 18,
0;" Hai colpito degli asteroidi
!! ": FOR g=1 TO 200: NEXT g:
LET m(y1,x1)=0: GO TO 1000
1110 IF m(y1,x1)<>2 THEN GO TO 130
0
1120 PRINT AT 18,0;"Hai colpito un in
crociatore "
1130 IF ener>=500 THEN LET m(y1,x1)=
0: LET incr=incr-1: PRINT AT 20,
0;"rimangono ";incr;" incociator
i"
1140 IF ener>=500 THEN IF incr<=0
THEN LET fla=2: FOR g=1 TO 150
: NEXT g: GO TO 9500
1150 IF ener<500 THEN PRINT AT 20,0;
"ma con energia insufficiente":
FOR g=1 TO 60: NEXT g: GO TO 50
00
1160 FOR g=1 TO 300: NEXT g: GO TO 10
00
1300 PRINT AT 18,0;"Hai colpito l'amm
iraglia "
1310 IF ener<1000 THEN PRINT AT 20,0
;"ma con energia insufficiente":
FOR g=1 TO 60: NEXT g: GO TO 55
00
1320 PRINT AT 20,0;" HAI COMPLETATO L
A MISSIONE "
1330 FOR g=1 TO 300: NEXT g: LET fla=
1: GO TO 9500
5005 GO SUB 7000: PRINT AT 14,18;"
ALLARME !!";AT 15,20;" missili i
n ";AT 16,18;" rotta di";AT 1
7,21;"collisione"
5007 FOR g=1 TO 30: BEEP .1,g: NEXT g
5010 IF RND<.65 THEN PAPER 7: CLS :
PAUSE 3: PAPER 0: CLS : FOR g=1
TO 70: NEXT g: LET en=en-(500+
INT (100*RND)): GO TO 1000

```



LOAD

Star war

```

5020 PRINT AT 20,0;"SIAMO STATI MANCA
TI ! DANNI 0 "
5025 FOR g=1 TO 300: NEXT g
5030 LET a=INT (RND*9+1): LET b=INT (
RND*9+1): IF m(a,b)<>0 THEN
GO TO 5030
5040 LET m(y1,x1)=0: LET m(a,b)=2:
GO TO 1000
5500 GO SUB 7000: PRINT AT 14,18;"
ALLARME !!";AT 15,20;" missili i
n ";AT 16,18;" rotta di";AT 1
7,21;"collisione"
5505 FOR g=1 TO 30: BEEP .1,g: NEXT g
5510 IF RND<.65 THEN PAPER 7: CLS :
PAUSE 3: PAPER 0: CLS : FOR g=1
TO 40: NEXT g: LET fla=3:
GO TO 9500
5520 PRINT AT 20,0;" SIAMO STATI MANC
ATI ! DANNI 0 ":
5525 FOR g=1 TO 300: NEXT g
5530 LET a=INT (RND*9+1): LET b=INT (
RND*9+1): IF m(a,b)<>0 THEN
GO TO 5530
5540 LET m(y1,x1)=0: LET m(a,b)=3:
GO TO 1000
5999 STOP
6000 REM {2SG8} DRAW comandi{2SG8}
6010 PAPER 1: INK 6: BORDER 1: CLS
6030 FOR g=16 TO 88 STEP 8: PLOT g,15
9: DRAW 0,-72: NEXT g
6040 FOR g=159 TO 87 STEP -8: PLOT 16
,g: DRAW 72,0: NEXT g
6050 PRINT AT 1,2;"abcdefghi"
6060 FOR g=1 TO 9: PRINT AT g+1,1;g:
FOR h=1 TO 9: IF m(g,h)<>0
THEN PRINT AT g+1,h+1;"{SG8}"
6070 NEXT h: NEXT g
6080 PRINT AT 12,1;"energia";AT 14,1;
"{7SG8}";AT 14,7-LEN STR$ en;
OVER 1;en
6090 PRINT AT 12,9;"mosse";AT 14,9;"
{5SG8}";AT 14,13-LEN STR$ mo;
OVER 1;mo
6100 PLOT 8,48: DRAW 112,0: DRAW 0,11
6
6300 RETURN
6500 REM {2SG8} RADAR {2SG8}
6510 INPUT "coordinate (riga/colonna)
?"; LINE i$
6520 IF LEN i$<>2 THEN GO TO 6510
6530 IF CODE i$(1)<50 OR CODE i$(1)>5
6 THEN PRINT AT 21,7;" RIGA I
NESATTA ": GO TO 6510
6531 IF CODE i$(2)<98 OR CODE i$(2)>1
04 THEN PRINT AT 21,7;"COLONNA
INESATTA ": GO TO 6510

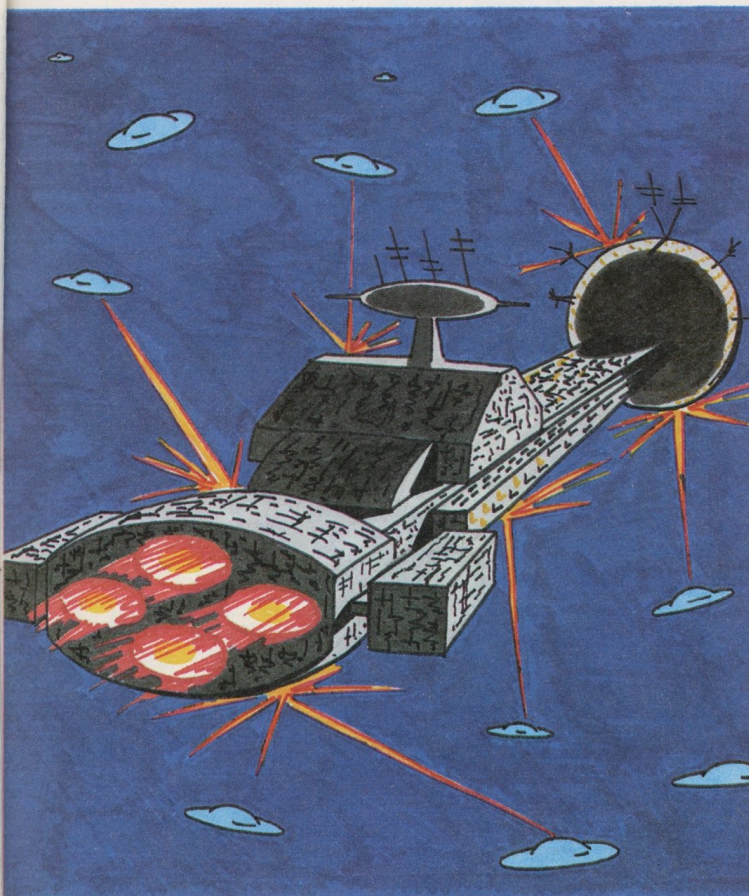
```

LOAD

Star war

```
6540 LET y1=VAL i$(1): LET x1=CODE i$(
(2))-96
6550 LET inc=0: LET met=0
6560 FOR g=-1 TO 1: FOR h=-1 TO 1:
PRINT AT (1+y1+h),(1+x1+g);
BRIGHT 1; PAPER 3; OVER 1;" ":
LET met=met+(m(y1+h,x1+g)=1):
LET inc=inc+(m(y1+h,x1+g)=2
OR m(y1+h,x1+g)=3): NEXT h:
NEXT g

6565 PRINT AT 21,0;l$
6570 GO SUB 7000:
6580 PRINT AT 14,18;"Qui sala radar"
6582 PRINT AT 15,18;"mi ricevete?"
6585 INPUT i$
6590 IF i$<>"roger" THEN GO TO 658
0
```



```
6595 PRINT AT 14,18;" Nel settore "
: PRINT AT 15,19;"ci sono: "

6597 PRINT AT 16,19;inc;" astronav"+(
"i" AND inc<>1)+("e" AND inc=1
)+" e";AT 17,19;met;" asteroid"+
("e" AND met=1)+("i" AND met
<>1)

6790 INPUT "ricevuto ?";i$
6800 IF i$<>"roger" THEN GO TO 659
5
6810 FOR g=0 TO 20: PRINT AT g,16;"
": NEXT g

6999 RETURN
7000 REM {2SG8} equipaggio{2SG8}
7001 PLOT 153,175: DRAW 0,-100:
DRAW 102,0: DRAW 0,100: DRAW -1
02,0
7005 PRINT AT 1,23;"{5SG3}"
7010 PRINT AT 2,21;" {SG2}{5SG8}
{SG1}"
7020 PRINT AT 3,21;" {5G3}{G7}{SG8}"
7030 PRINT AT 4,22;" {SG8}"
7040 PRINT AT 5,28;"{SG8}"
7041 PRINT AT 6,28;"{G5}"
7050 PLOT 180,152: DRAW 0,-24: DRAW 5
,-13: DRAW 1,-10: DRAW 5,-10:
DRAW 10,-1

7060 DRAW 10,1: DRAW 8,13
7070 PLOT 227,128: DRAW -2,-38:
DRAW -14,-13: DRAW -15,-1:
DRAW -2,17
7080 PLOT 199,128: DRAW -2,-10:
DRAW 2,-1: PLOT 202,118: DRAW -
1,2
7090 PLOT 208,136: DRAW 3,0: PLOT 209
,135
7091 PLOT 190,136: DRAW 3,0: PLOT 191
,135
7110 PLOT 225,90: DRAW 27,-7: DRAW 3,
-6
7111 PLOT 193,89: DRAW -33,-7: DRAW -
3,-6

7112 PLOT 194,104: DRAW 10,0,1.2:
DRAW -10,0,1.2
7120 RETURN
9000 PRINT " Hai esaurito l'erner
gia prima di completare lamis
sione ": GO TO 9990
9500 IF fla=3 THEN GO TO 9550
9505 CLS : PRINT " HAI COMPLETATO L
A MISSIONE "
9510 PRINT " IN ";mo;" MOSS
E"
9530 GO TO 9990
9550 PRINT "SEI STATO COLPITO PRIMA
DI COM- PLETARE LA MISSIONE"
9990 INPUT "UN ALTRA PARTITA (S/N)";i
$: IF i$="s" THEN RUN
```

Log hop

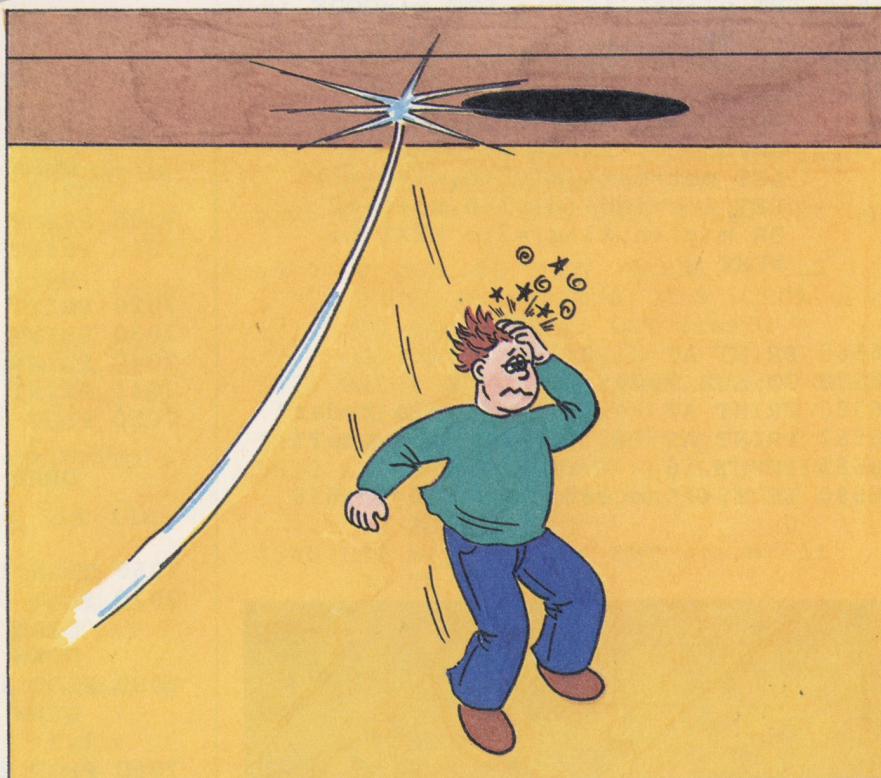
di **Christopher Gibbs**
trad. e adatt.
di **Andrea Violini**

**Un programma breve, veloce, divertente e impegnativo.
Più di così...**

Lo scopo del gioco è quello di far saltare un omino attraverso le aperture presenti nelle piattaforme scorrevoli, per raggiungere gli anelli posti in cima allo schermo.

L'omino viene controllato per mezzo dei tasti 5 e 8 (per muoverlo rispettivamente a sinistra e a destra) e dello 0 (per saltare). Se durante un salto l'omino manca l'apertura e sbatte la testa contro la piattaforma soprastante, rimarrà stordito per un attimo, rischiando di cadere, attraverso aperture successive, sempre più in basso.

Sono inclusi 3 livelli di difficoltà e i record vengono registrati con il nome del giocatore autore della performance (non conosciamo, al momento, punteggi superiori al 93%). Il punteggio è funzione del tempo impiegato per raggiungere gli anelli; bisognerà essere quindi capaci di saltare a colpo sicuro nel più breve tempo possibile. Attenti ai ber-noccoli!



Descrizione del programma

Linee

30-100
200-250
300-420

1000-1080

1100-1120

8000-8110

9000-9210

9500-9565

9600-9650

Commento

routine principale

routine per la caduta

routine per la morte, stima del punteggio, aggiornamento della tabella dei record

routine per il salto

raggiungimento degli anelli

presentazione e istruzioni

grafici (UDG)

disposizione casuale delle piattaforme

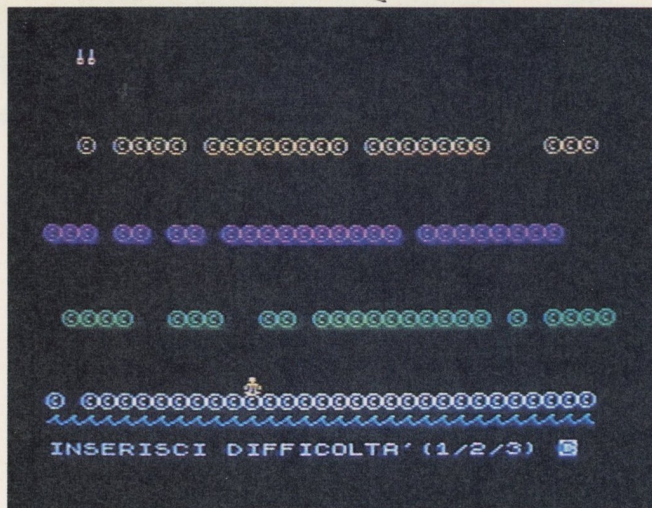
preparazione dello schermo, inizio del gioco

Log hop

versione per ZX Spectrum 16K/48K

```
2 POKE 23658,8
3 PAPER 0: INK 7: BORDER 0: CLS
5 LET HOLE=0: LET game=0: DIM z(10)
  ): DIM f$(10,3)
```

```
10 GO SUB 8000
12 LET SCORE=1000: LET dead=0
13 LET game=1
15 LET x=19: LET y=10
20 LET e$="A"
30 FOR m=31 TO 1 STEP -1
31 IF SCORE>1 THEN LET SCORE=SCORE
  -2
```



```

32 IF dead>0 THEN LET dead=dead-1:
    BEEP .05,-10
40 PRINT INK 6;AT 5,0;a$(32-m TO )
    ;a$( TO 32-m);AT 10,0; INK 3;b$(
    m TO );b$( TO m);AT 15,0; INK 4;
    c$(32-m TO );c$( TO 32-m); INK 7
    ;AT 20,0;d$(m TO );d$( TO m)
43 IF dead>0 THEN GO TO 55
45 IF m/2=INT (m/2) THEN LET e$="B
    "
46 IF m/3=INT (m/3) THEN LET e$="K
    "
50 PRINT INK 6;AT x,y;" ";e$;" "
53 IF m/3=INT (m/3) THEN GO TO 57
55 IF SCREEN$(x+1,y+1)=" " THEN
    GO SUB 200
57 IF dead>0 THEN GO TO 100
60 LET y=y+(INKEY$="8")-(INKEY$="5"
    )
85 IF y<0 THEN LET y=29: PRINT
    AT x,0;" "
86 IF y>29 THEN LET y=0: PRINT
    AT x,29;" "
90 IF INKEY$="0" THEN GO SUB 1000
95 LET e$="A"
100 INK 7: NEXT m: GO TO 30
200 REM CADUTA
205 INK 6
210 IF x=19 THEN GO TO 300
220 FOR n=x+1 TO x+5: PRINT AT n,y+1
    ;"H";AT n-1,y+1;" ": BEEP .05,20
    -n: NEXT n
230 LET e$="E": LET x=n-1: PRINT
    AT x,y+1;e$: LET dead=D*2
240 IF SCREEN$(x+1,y+1)=" " THEN
    GO TO 210
250 RETURN
290 REM MORTE
300 PRINT AT 19,y+1;"H": PAUSE 2:
    PRINT AT 20,y+1;"H";AT 19,y+1;"
    "
305 BEEP .1,0
310 PRINT AT 20,y+1;" "
320 FOR n=y+1 TO 30: PRINT INK 5;
    AT 21,n;"E"; INK 6;"E";AT 21,n;
    INK 5;"F": BEEP .05,30-n:

```

```

NEXT n: PRINT INK 5; BRIGHT 1;
    AT 21,31; INK 5;"F"
321 FOR n=1 TO 10: NEXT n: LET SCORE
    =0: GO TO 330
325 LET SCORE=INT (100*(SCORE/1000))
330 PRINT AT 2,10;" ";AT 3,
    10;" ";AT 4,10;"
    ";AT 3,10;"PUNTI=";SCORE;"%"
340 PRINT AT 17,3; FLASH 1;"PREMI EN
    TER PER CONTINUARE": IF CODE
    INKEY$<>13 THEN GO TO 340
345 CLS
350 FOR n=1 TO 10: IF SCORE>z(n)
    THEN GO TO 400
360 NEXT n
370 PRINT AT 0,4;"I GRANDI DI OGGI":
    FOR n=1 TO 9: PRINT AT n*2,6;n;
    "=";" ";f$(n);" ... ";z(n):
    NEXT n: PRINT AT 20,5;10;"=" ;f
    $(10);" ... ";z(10)
380 GO TO 9515
400 FOR m=10 TO n+1 STEP -1: LET z(m)
    =z(m-1): LET f$(m)=f$(m-1):
    NEXT m: LET z(n)=SCORE
410 INPUT "INSERISCI LE TUE INIZIALI
    (MAX 3) ";g$: IF LEN g$>3
    THEN GO TO 410
420 LET f$(n)=g$: GO TO 370
1000 REM SALTO
1005 INK 6
1007 IF x=4 AND y+1=po THEN GO TO 11
    00
1010 IF SCREEN$(x-4,y+1)="@" THEN
    LET dead=D*2: GO TO 1050
1020 PRINT AT x,y;" C ": BEEP .1,10:
    FOR n=x-1 TO x-5 STEP -1:
    PRINT AT n,y+1;"D";AT n+1,y+1;"
    ": BEEP .05,20-n: NEXT n
1025 LET x=n+1
1030 RETURN
1050 PRINT AT x,y+1;"C": PAUSE 2:
    FOR n=x-1 TO x-3 STEP -1:
    PRINT AT n,y+1;"D";AT n+1,y+1;"
    ": BEEP .05,20-n: NEXT n
1055 PRINT INK 7;AT n+1,y+1;"D"
1060 BEEP .1,10
1070 FOR k=n+2 TO x: PRINT AT k,y+1;"
    H";AT k-1,y+1;" ": BEEP .05,20-k
    : NEXT k
1072 PRINT INK 6;AT k-1,y+1;"E"
1075 LET e$="E"
1080 RETURN
1100 REM FINE
1110 PRINT AT x,y+1;"C": PAUSE 2:
    FOR n=x-1 TO x-3 STEP -1:
    PRINT AT n,y+1;"D";AT n+1,y+1;"
    ": BEEP .05,20-n: NEXT n
1115 PRINT AT 0,PO;"I";AT 1,PO;"J":
    FOR N=1 TO 10: BEEP .1,N:
    NEXT N: FOR N=20 TO 0 STEP -2:
    BEEP .1,N: NEXT N
1120 GO TO 325
8010 LET m$="@   @   @   @   @
    @   @   @   @   @
    @   @   @   @   @

```

```

8012 LET n$=" "
8014 LET o$=" "
8016 LET p$=" "
8017 LET q$=" "

8020 FOR m=27 TO 2 STEP -1
8025 BEEP .01,m
8030 PRINT INK 2;AT 4,m;m$( TO 28-m)
; INK 3;AT 5,m;n$( TO 28-m);
INK 4;AT 6,m;o$( TO 28-m);
INK 5;AT 7,m;p$( TO 28-m);
INK 6;AT 8,m;q$( TO 28-m)
8040 PAUSE 2: NEXT m
8045 FOR j=1 TO 3
8050 FOR n=1 TO 6: INK n: PRINT AT 4,
2;m$;AT 5,2;n$;AT 6,2;o$;AT 7,2;
p$;AT 8,2;q$
8055 BEEP .01,n+10+(j*2): NEXT n
8056 NEXT j
8060 PRINT AT 12,6; INK 7; BRIGHT 1;"
DI C.M.GIBBS 1983";AT 13,0;"TRA
D. E ADATT. DI A.VIOLINI 1984"
8070 PAUSE 100
8080 DIM S$(32): PRINT AT 12,0;S$;
AT 13,0;S$
8090 REM REGOLE
8100 PRINT AT 11,2; INK 7; BRIGHT 1;"
5: SINISTRA";AT 13,2;"8: DESTRA"
;AT 15,2;"0: SALTO";AT 17,0;"EVI
TA I TRABOCCHETTI,""RAGGIUNGI G
LI ANELLI""E NON FARTI ABBATTER
E !!!"
8110 PRINT FLASH 1;AT 21,5;"ATTENDI
UN ATTIMO"
9000 REM UDG
9010 FOR i=1 TO 11: FOR n=0 TO 7
9015 READ a: POKE USR CHR$(143+i)+n,
a: NEXT n: NEXT i
9020 DATA 28,28,8,63,40,14,82,35
9040 DATA 28,28,9,254,24,232,136,12
9060 DATA 68,186,185,82,60,199,66,36
9080 DATA 28,28,8,62,85,85,20,119
9100 DATA 0,6,2,18,18,210,222,254
9115 DATA 0,0,0,12,24,40,196,3
9130 DATA 66,66,66,66,231,165,231,0
9150 DATA 195,36,24,146,254,16,56,56
9170 DATA 66,66,66,66,231,231,231,130

9190 DATA 186,186,254,16,56,68,68,198

9210 DATA 28,28,255,8,15,241,0,0
9500 REM STRINGHE
9510 DIM a$(31): DIM b$(31): DIM c$(3
1): DIM d$(31)
9515 IF game=1 THEN LET b=1+INT (
RND*3): GO TO 9520
9517 FOR b=1 TO 3
9520 FOR n=1 TO 31
9530 LET r=INT (RND*6)
9540 IF r=2 THEN LET HOLE=1:
GO SUB 9548
9545 IF b=1 THEN LET a$(n)=" ":
GO TO 9551

```



```

9546 IF b=2 THEN LET b$(n)=" ":
GO TO 9551
9547 IF b=3 THEN LET c$(n)=" ":
GO TO 9551
9548 IF b=1 THEN LET a$(n)=" "
9549 IF b=2 THEN LET b$(n)=" "
9550 IF b=3 THEN LET c$(n)=" "
9551 NEXT n
9552 IF HOLE=0 THEN LET A$(1)=" ":
LET B$(1)=" ": LET C$(1)=" "
9553 LET HOLE=0
9555 IF game=1 THEN CLS : GO TO 9600

9560 NEXT b
9565 LET d$=" "
9570 PRINT AT 21,3; FLASH 0;" PRE
MI UN TASTO ": PAUSE 0
9575 CLS
9600 LET Z$="F": FOR M=31 TO 1 STEP -
1
9610 PRINT INK 6;AT 5,M;a$( TO 32-M)
;AT 10,0; INK 3;b$(m TO );AT 15,
M; INK 4;c$( TO 32-M); INK 7;
AT 20,0;d$(m TO )
9615 PRINT INK 5;AT 21,0;Z$
9616 LET Z$=Z$+"F"
9620 NEXT M
9625 PRINT AT 19,11;"D"
9630 LET po=2+INT (RND*27): PRINT
AT 0,po; INK 6; BRIGHT 1;"G"
9640 INPUT "INSERISCI DIFFICOLTÀ'(1/2
/3) ";D
9645 IF d<>INT d OR d<1 OR d>3
THEN GO TO 9640
9650 GO TO 12..

```



Memogame

di Sean Ghisoni

La memoria visiva è una dote spesso non sufficientemente allenata...

Questo programma si rifà ad un classico gioco didattico di memoria visiva. Il computer distribuisce in modo casuale su di una scacchiera 16 coppie di carte uguali, che vengono poi coperte: i giocatori (o il giocatore) dovranno scoprire le carte a due a due, cercando di indovinare le coppie uguali; in caso di esito positivo guadagneranno un punto, mantenendo il controllo del gioco, e le carte rimarranno scoperte; in caso contrario il turno cambierà e le carte verranno ricoperte. Il programma è stato pensato per poter permettere a chiunque di giocare, dai 6 anni in avanti, compresi coloro che non hanno dimestichezza con il computer e la sua tastiera. Per questo il programma comprende un buon numero di sicure, atte ad impedire un suo blocco involontario; per lo stesso motivo si è preferito usare i numeri sulla scacchiera sia per il riferimento alle coordinate verticali sia per quelle orizzontali.

In questo modo la ricerca dei tasti da premere è facilitata dalla loro disposizione, mentre si era dimostrata in precedenza più difficoltosa la ricerca di lettere da parte di chi non aveva una sufficiente conoscenza della tastiera. Il meccanismo del gioco è facile da apprendere e le istruzioni contenute nel programma mettono in grado chiunque di poterlo usare. Durante lo svolgimento della partita il computer stesso indicherà il turno di gioco, la coordinata che va introdotta ed eventuali errori (nel caso le coordinate si riferiscano a carte già scoperte). Da prove fatte si è verificato che anche bambini di 6/7 anni non hanno avuto difficoltà nella comprensione del gioco, mentre per loro è stato più difficile capire la dinamica del meccanismo di utilizzazione delle coordinate per determinare la posizione della carta scelta; tale problema, in ogni caso, si risolve dopo mezz'ora od un ora al massimo di gioco assistito da un adulto e permette tra l'altro al bambino di apprendere, sia pur in modo empirico, ma sicuramente



piacevole, il concetto di coordinate cartesiane. Anche il gradimento è stato buono, sia da parte di bambini che di adulti, pur non essendo un gioco di movimento. Alcuni si sono però lamentati che il tempo a disposizione per memorizzare le carte, 10 secondi circa a partire da quando il computer ha terminato la disposizione delle stesse, è troppo poco; a tale proposito si precisa che questa è già una concessione extra, in quanto la versione da tavolo originale prevede che si inizi il gioco con tutte le carte coperte e che la memorizzazione avvenga ricordando il soggetto e la posizione delle coppie spaiate che vengono girate durante la partita stessa. In ogni caso non è difficile adattare questo tempo ai propri gusti personali: per aumentarlo

basta infatti intervenire sulla linea 8100, allungando il ciclo FOR...NEXT, oppure aumentando la durata dei BEEP alla linea 8120; va da sé che per diminuire il tempo si procederà a diminuire tali valori. È stato scelto un ciclo FOR...NEXT in sostituzione a PAUSE x per impedire che la pressione casuale di un tasto accorciasse prematuramente il tempo a disposizione.

Chi vorrà invece iniziare senza la visualizzazione iniziale delle carte non dovrà far altro che eliminare le linee 8110, 8120 e 8130 e sostituire la linea 8080 con la seguente: 8080 PRINT AT 01,v1;PAPER 0;CHR\$ 143. In questo modo, all'inizio il computer visualizzerà sulla scacchiera dei quadratini neri, anziché le carte. Va ancora detto che, se

non piacesse tutti o in parte i caratteri grafici attualmente inseriti nel programma, essi potranno essere cambiati, sostituendo a piacere le linee DATA presenti, escluse la 9260 e la 9270, che servono per il disegno della scacchiera.

Una volta copiato il programma si darà un RUN 2500 per la registrazione su nastro dello stesso e relativa verifica. Come noterete, in questa linea vi è anche un comando di autostart alla linea 8500; pertanto, se a questo punto vorrete iniziare subito a giocare, dovrete dare un RUN 8500, cosa che non sarà più necessaria quando il programma verrà caricato da nastro. Ora non rimane altro da fare che augurarvi buon lavoro, e perché no, buon divertimento!

Descrizione del programma

Linee

990-1390

Commento

Regolano lo svolgimento del gioco, indicano il turno di gioco, le coordinate da introdurre, scoprono ed eventualmente ricoprono le carte, aggiornano il punteggio e controllano se la partita è terminata o no.

1998-2270

Fine partita. Se il giocatore è uno solo, indicano il numero di mosse fatte per terminare. Chiedono se si vuol giocare ancora: in caso di risposta negativa provocano uno stop, altrimenti chiedono se cambiano o no i giocatori.

2500

Salvataggio e verifica su nastro del programma.

6998-7350

Inizio partita: chiedono quanti sono i giocatori e i loro nomi, inizializzano lo schermo del gioco.

7998-8300

Stabiliscono casualmente la posizione delle carte sulla scacchiera, visualizzandole, memorizzano le loro coordinate, danno il tempo ai giocatori di memorizzarle per poi ricoprirle.

8498-8670

Presentazione e istruzioni del gioco.

8998-9070

Caricano i caratteri grafici delle linee DATA.

9080-9270

Linee DATA dei caratteri grafici.

Memogame

versione per ZX Spectrum 16K/48K

```

990 REM *****
    *      MEMOGAME      *
    * ©84 sean ghisoni *
    *****

995 REM
998 REM **SVOLGIMENTO PARTITA**
999 REM {23G3}
1000 LET g2=0: LET cf=0: LET p1=0:
    LET p2=0: LET p3=0: LET p4=0:
    LET nm=0
1010 PRINT AT 17,24;"GIOCA:"
1020 FOR i=1 TO ng
1025 LET p=0
1040 IF i=1 THEN LET x$b$: LET p=p1
1050 IF i=2 THEN LET x$c$: LET p=p2
1060 IF i=3 THEN LET x$d$: LET p=p3
1070 IF i=4 THEN LET x$e$: LET p=p4
1080 FOR n=1 TO 2
1085 IF cf=18 THEN GO TO 2000
1090 PRINT AT 18,24;"      "
1100 PRINT BRIGHT 1;AT 18,24;x$
1110 PRINT FLASH 1;AT 20,24;"C.ORIZ.
    ?"
1120 IF INKEY$<>"" THEN GO TO 1120

1130 IF INKEY$="" THEN GO TO 1130
1150 LET j$=INKEY$: IF CODE j$<49
    OR CODE j$>54 THEN GO TO 1120
1160 LET o=VAL j$
1170 LET ol=o*3+1
1180 PRINT BRIGHT 1; OVER 1;AT ol,1;
    " "
1190 PRINT FLASH 1;AT 20,24;"C.VERT.
    ?"

```

```

1200 IF INKEY$<>"" THEN GO TO 1200
1210 IF INKEY$="" THEN GO TO 1210
1220 LET j$=INKEY$: IF CODE j$<49
    OR CODE j$>54 THEN GO TO 1200
1240 LET v=VAL j$
1250 LET vl=v*3
1260 PRINT OVER 1;AT ol,1;" "
1270 IF ATTR (ol,vl)<>56 THEN
    BEEP 1,1: GO TO 1110
1280 LET g=c(o,v): LET car=g-144:
    IF g>151 THEN LET car=car-8
1290 BEEP .05,0: PRINT BRIGHT 1;
    INK 9; PAPER car;AT ol-1,vl;"
    ";AT ol,vl;" ";CHR$ g;" ";AT ol
    +1,vl;" "
1300 IF ng=1 AND n=2 THEN LET nm=nm+
    1
1310 IF n=1 THEN LET gl=g: LET o2=ol
    : LET v2=vl
1320 IF n=2 THEN LET g2=g: LET o3=ol
    : LET v3=vl
1330 IF n=2 AND gl=g2 THEN LET p=p+1
    : PRINT AT i*3+1,30;p: LET cf=cf
    +1: GO TO 1080
1340 IF n=2 AND gl<>g2 THEN BEEP 1
    ,-15: PRINT AT o2-1,v2;" T ";
    AT o2,v2;"S S";AT o2+1,v2;" T ";
    AT o3-1,v3;" T ";AT o3,v3;"S S";
    AT o3+1,v3;" T "
1350 IF i=1 THEN LET p1=p
1360 IF i=2 THEN LET p2=p
1370 IF i=3 THEN LET p3=p
1380 IF i=4 THEN LET p4=p
1390 NEXT n: NEXT i: GO TO 1020
1998 REM **FINALE**
1999 REM {10G3}
2000 FOR b=1 TO 25

```



```

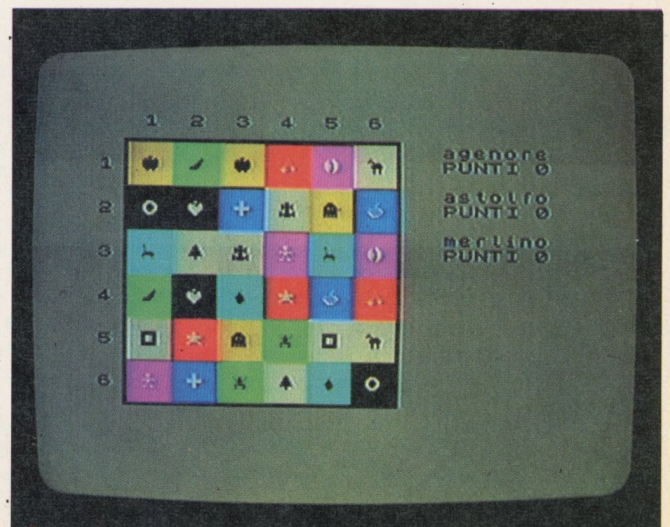
2010 BEEP .05,INT (RND*20)+30
2020 NEXT b
2030 FOR n=3 TO 20
2040 PRINT AT n,3;"
"
2050 NEXT n
2060 IF ng=1 THEN PRINT AT 10,7;"HAI
CHIUSO"
2070 IF ng=1 THEN PRINT AT 12,5;"LA
PARTITA IN:"
2080 IF ng=1 THEN PRINT AT 14,8;nm;"
MOSSE": FOR n=1 TO 10: BEEP .05
,n-5: BEEP .05,n+5: NEXT n
2085 FOR n=1 TO 30: BEEP .05,n-5:
BEEP .05,n+5: NEXT n
2090 FOR n=3 TO 20
2100 PRINT AT n,3;"
"
2110 NEXT n
2120 PRINT AT 8,8;"UN'ALTRA"
2130 PRINT AT 10,8;"PARTITA?"
2140 PRINT AT 12,6;"SE SI PREMI:"
2150 PRINT FLASH 1;AT 13,11;"S"
2160 PRINT AT 14,5;"UN ALTRO TASTO"
2170 PRINT AT 15,6;"PER SMETTERE"
2180 IF INKEY$="" THEN GO TO 2180
2190 IF INKEY$<>"s" THEN CLS :
PRINT AT 10,15;"OK": FOR n=0
TO 50: BEEP .02,n: NEXT n:
STOP
2200 FOR n=3 TO 20
2210 PRINT AT n,3;"
"
2220 NEXT n
2230 PRINT AT 10,7;"CAMBIANO I"
2240 PRINT AT 11,4;"GIOCATORI (S/N)?"
2250 IF INKEY$<>" " THEN GO TO 2250
2260 IF INKEY$="" THEN GO TO 2260
2270 GO TO 8000-(1000 AND INKEY$="s")
2500 SAVE "MEMOGAME" LINE 8500:
VERIFY "": STOP
6998 REM **INIZIO PARTITA**
6999 REM {18G3}
7000 CLS : PRINT AT 20,0;"QUANTI GIOC
ATORI ? ( da 1 a 4 )"
7010 INPUT ;ng
7020 IF ng<1 OR ng>4 THEN GO TO 7010
7030 FOR i=1 TO ng
7040 PRINT AT 20,0;" NOME DEL GIOCA
TORE NUMERO :";i
7050 INPUT LINE g$: IF LEN g$>8
THEN GO SUB 7050
7060 IF i=1 THEN LET b$=g$
7070 IF i=2 THEN LET c$=g$
7080 IF i=3 THEN LET d$=g$
7090 IF i=4 THEN LET e$=g$
7100 BEEP .05,0: PRINT AT i*3,24;g$
7110 PRINT AT i*3+1,24;"PUNTI 0"
7120 NEXT i
7130 PRINT AT 20,0;"
"

```

```

7200 PRINT AT 1,4;"1 2 3 4 5 6"
7210 LET z=1
7220 FOR n=4 TO 19 STEP 3
7230 PRINT AT n,1;z
7240 LET z=z+1
7250 NEXT n
7260 PLOT 23,152
7270 DRAW 145,0
7280 DRAW 0,-145
7290 DRAW -145,0
7300 DRAW 0,145
7310 PLOT 22,153
7320 DRAW 147,0
7330 DRAW 0,-147
7340 DRAW -147,0
7350 DRAW 0,147
7998 REM **POSIZIONE CARTE**
7999 REM {19G3}
8000 DIM c(6,6)
8005 RANDOMIZE 0
8010 FOR n=1 TO 2: FOR i=1 TO 18
8020 LET g=144
8025 FOR i=1 TO 18
8030 LET o=1+INT (RND*6)
8040 LET v=1+INT (RND*6)
8050 LET ol=o*3+1: LET vl=v*3
8060 IF ATTR (ol,v1)<>56 THEN
GO TO 8030
8065 LET c(o,v)=g
8070 LET car=g-144: IF g>151 THEN
LET car=car-8
8080 PRINT PAPER car; INK 9;
BRIGHT 1;AT ol-1,v1;" ";AT ol
,v1;" ";CHR$ g;" ";AT ol+1,v1;"
"
8090 LET g=g+1
8100 NEXT i: NEXT n
8110 FOR b=1 TO 15
8120 BEEP .25,30: BEEP .25,50
8130 NEXT b
8140 FOR n=4 TO 21 STEP 3
8150 PRINT AT n-1,3;" T T T T T
T "
8160 PRINT AT n+1,3;" T T T T T
T "
8170 PRINT AT n,3;"S SS SS SS SS SS S
"

```



LOAD

Memogame

```
8180 NEXT n
8190 FOR n=1 TO ng
8200 PRINT AT n*3+1,30;"0 "
8210 NEXT n
8300 GO TO 1000
8498 REM **ISTRUZIONI**
8499 REM {14G3}
8500 CLS : PRINT AT 10,7; FLASH 1;"FERMARE IL NASTRO": BEEP 1,1:
      PAUSE 200
8510 BORDER 7: PAPER 7: CLS
8520 PRINT TAB 9;"***MEMOGAME***"
8530 PRINT TAB 9;"{14G3}"
8540 PRINT TAB 5;"GIOCO DI MEMORIA VISIVA"
8550 FOR b=1 TO 10: BEEP .05,30-b:
      BEEP .05,30+b: NEXT b
8560 PRINT "All'inizio del gioco, su di una scacchiera di 36 caselle, vengono presentate per circa 10 secondi 16 coppie di carte."
8570 PRINT "Il gioco consiste nel fare il maggior numero di coppie. Le carte vengono scoperte introducendo le relative coordinate della scacchiera."
8580 PRINT "In caso le due carte scoperte siano uguali, il giocatore guadagna 1 punto e continua il gioco. Altrimenti esse vengono ricoperte ed il turno passa ad un altro giocatore."
8590 PRINT ""PREMI UN TASTO PER CONTINUARE"
8600 PAUSE 0: CLS
8610 PRINT "Durante il gioco un BEEP di tonobasso indica che le due carte sono diverse, un BEEP acuto indica che si e' cercato di scoprire una carta gia' scoperta."
8620 PRINT "I giocatori possono essere da 1 a 4; all'inizio del gioco viene richiesto il loro numero ed i loro nomi (numero/nomi seguiti dal tasto ENTER). Per le carte basta premere i tasti delle relative coordinate."
8630 PRINT "Durante il gioco vengono indicati i punteggi dei giocatori, il turno di gioco e la coordinata che deve essere introdotta. In caso si giochi da soli, a fine partita viene visualizzato anche il numero di mosse effettuate."
```

```
8650 PRINT ""PREMI UN TASTO PER INIZIARE"
8660 PAUSE 0
8670 CLS : PRINT FLASH 1;AT 10,10;"ATTENDERE"
8998 REM **CARATTERI GRAFICI**
8999 REM {21G3}
9010 FOR i=144 TO 163
9020 FOR n=0 TO 7
9030 READ a
9040 POKE USR CHR$(i+n,a)
9050 NEXT n
9060 NEXT i
9070 GO TO 7000
9080 DATA 24,82,247,255,255,126,60,24
9090 DATA 4,8,8,86,171,213,106,60
9100 DATA 8,8,20,20,34,99,243,96
9110 DATA 44,110,231,247,247,247,102,44
9120 DATA 2,3,7,14,30,124,248,0
9130 DATA 8,16,24,60,124,62,60,24
9140 DATA 24,82,255,255,255,255,126,36
9150 DATA 255,255,195,195,195,195,255,255
9160 DATA 60,126,231,195,195,231,126,60
9170 DATA 24,24,24,255,255,24,24,24
9180 DATA 24,24,255,255,60,126,195,129
9190 DATA 56,56,16,254,186,40,40,238
9200 DATA 66,153,90,60,24,126,66,102
9210 DATA 192,64,64,64,124,254,132,132
9220 DATA 56,124,214,214,254,254,170,170
9230 DATA 8,28,28,62,62,127,8,8
9240 DATA 64,160,30,63,63,18,18,18
9250 DATA 24,90,126,90,24,153,255,189
9260 DATA 0,0,0,255,255,0,0,0
9270 DATA 24,24,24,24,24,24,24,24
```



For Love

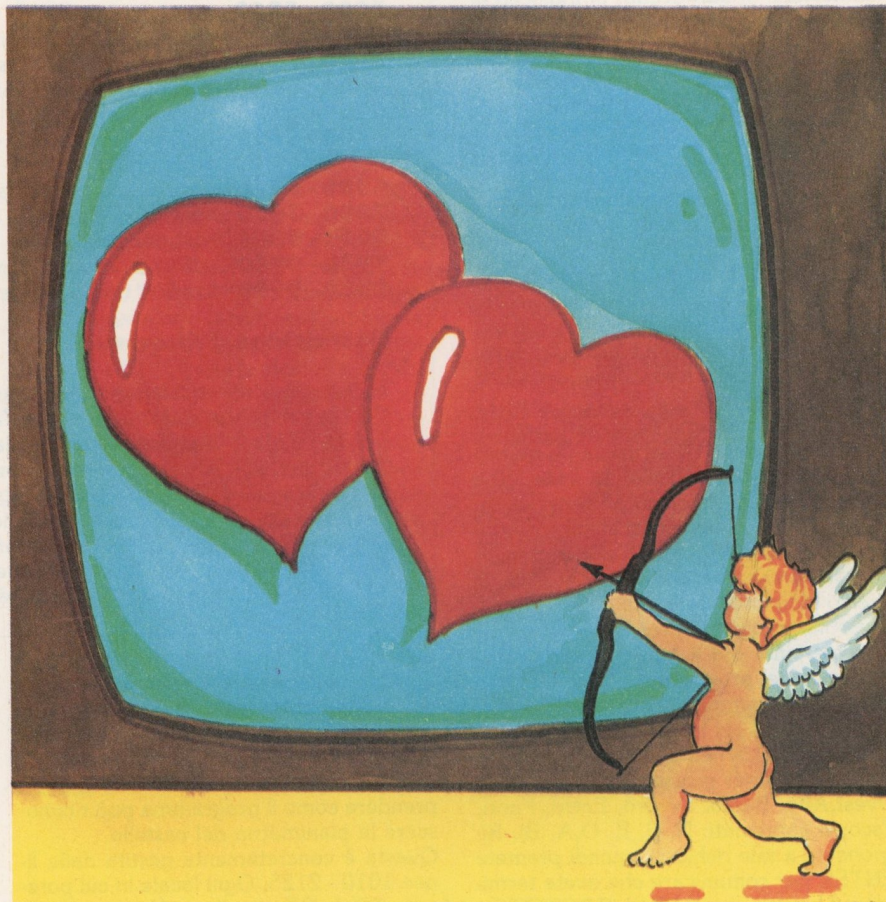
di Mark Chapman

trad. e adatt. di
Carlo Panzalis**Di questi tempi, per un giovane innamorato riuscire a giungere al matrimonio è una vera e propria ... avventura!**

Cosa non si fa per amore... In questa affascinante adventure game recitate la parte di un aitante giovane, disposto a penetrare all'interno della "Casa del Destino" per affrontare le innumerevoli insidie che il potente padrone di casa ha posto a difesa della sua privacy e dei suoi tesori. Siete quindi pronti a dimostrare all'adorata fidanzata il vostro coraggio ed amore e al di lei augusto e ricco genitore il vostro completo disinteresse per la dimensione materiale e venale della vita, assicurandolo così sulla sorte del suo imponente patrimonio.

Il programma qui pubblicato non potrà non rallegrare gli amanti del genere adventure, non solo perché il gioco è realmente avvincente e di difficilissima soluzione, ma anche perché costituisce senz'altro ottimo materiale, suscettibile di approfondito esame da parte di quanti volessero espandere la propria produzione artigianale a questo tipo di giochi. In questo scritto non esamineremo in profondità le caratteristiche operative del programma: correremo il rischio di privarvi del piacere di scoprire da soli i misteri di questa ostile magione. Ci limiteremo dunque a descrivere le modalità di gioco e ad esaminare il listato "a blocchi". All'inizio del gioco vi trovate all'interno del palazzo, di fronte al portone d'ingresso, dal quale è subito chiaro che... non potrete più uscire! Scoprire la strada d'uscita è infatti scopo primario del gioco. Attenzione, però: uscire senza aver trovato e raccolto il lingotto d'oro e il prezioso rubino, celati chissà dove nel palazzo, significa rinunciare a Elena (...la vostra amata: a proposito, chi intende fare cosa gradita alla sua ragazza farà bene a cambiare il nome nel listato... poi non venite a lamentarvi con noi!) e deludere una immensa folla che, informata della vostra impresa, si raduna all'uscita della casa, pronta ad acclamarvi in caso di successo.

Per realizzare il vostro scopo avete circa quaranta minuti di tempo, a meno che



riuscite a scovare e a mangiare la Frutta della Vita, magico alimento che rende immortale chi ha la fortuna di cibarsi di esso. Abbiamo così indicato i quattro obiettivi fondamentali del gioco, possibilmente da perseguire nel seguente ordine: frutta, lingotto e rubino, uscita. E' giusto comunque sottolineare che il pasto a base di frutta non è un imperativo categorico, ma solo un mezzo per avere un po' di respiro. È cioè possibile vincere la partita senza essersi curati di questo aspetto. Nulla abbiamo invece ancora detto sulle numerose insidie, trabocchetti ed incontri letali che si oppongono alla riuscita dell'impresa: la stanza degli scheletri, quella della vedova nera, il gas velenoso che si diffonde di colpo fra le stanze, ecc.. Anche in questo caso non intendiamo però sbilanciarci, rivelandovi cose che non do-

vete conoscere per giocare la partita in tutta onestà. D'altro canto tali insidie sono pressoché ineliminabili nel senso che, una volta incontrate, non vi lasciano scampo. Bisogna dunque munirsi di santa pazienza e ricordarsi da una partita all'altra quali stanze sono assolutamente da evitare. Ricordate che questi pericoli sono di tale entità da vanificare anche il dono dell'immortalità conquistato con l'ingerimento della Frutta della Vita.

Ciò che invece vi ostacolerà in ogni partita è la complicata planimetria del palazzo, unitamente alla necessità di raccogliere e assemblare i più strani oggetti che troverete sparsi in ogni dove, che vi consentiranno di ottenere fondamentali suggerimenti per accedere a locali altrimenti non raggiungibili e trovare così ciò che cercate. Chi è impaziente di trovare

l'uscita potrà, esaminando l'elenco degli oggetti presenti nel palazzo, scoprire le relazioni fra di essi, per risolvere più rapidamente il gioco. Siamo tuttavia certi che, anche in questo caso, non riuscirete a scoprire interamente le modalità da seguire per guadagnare rapidamente l'uscita con i premi in vostro possesso.

Chiariti scopi e problemi del gioco, passiamo ad esaminare le caratteristiche operative del programma. Il listato può essere suddiviso in tre blocchi principali: a) le routine di controllo, che si occupano della normale gestione del gioco, della verifica dei vostri comandi, dell'apparizione dei messaggi per informarvi circa il manifestarsi di dati eventi; b) il blocco di immagazzinamento della pianta del palazzo, che tiene dunque conto della disposizione delle stanze nello spazio e regola le possibilità di accesso fra un locale e quelli contigui; c) infine il blocco per la gestione delle situazioni particolari, cui si accede in relazione a vostre mosse particolari (raccolta o assemblamento di premi, scoperta di passaggi segreti, azioni bizzarre, ecc.).

Diversamente da quanto siamo soliti fare, la suddivisione del listato in questi tre blocchi è effettuata considerando, più che la semplice sequenzialità fisica fra linee di programma, la loro consequenzialità logica ed operativa.

In tal senso consideriamo appartenenti al medesimo gruppo di istruzioni (in questo caso routine di controllo) le linee 40 - 330, che sovraintendono all'accoglimento degli input da tastiera e alla loro verifica, e le linee 9000 - 9050, che contengono i dati atti a consentire tale verifica, e questo indipendentemente dalla loro collocazione in distinte aree del listato.

Consideriamo ora il funzionamento vero e proprio del programma. Una volta lanciato, il programma vi richiede di inserire la direzione in cui intendete muovervi. A tale scopo disponete di un menu di 16 comandi (10 per esteso: Nord, Sud, Est, Ovest, Alto, Basso, Dentro, Entro, Fuori, Esco; 6 abbreviati: N, S, E, O, A, B). Rispondete a tale richiesta, quindi premete ENTER per comunicare che avete terminato il vostro comando. Le linee 100 - 190 si occupano dell'accoglimento di questo input; le linee successive (200 - 330) si occupano invece della verifica della correttezza del comando sia per quanto riguarda la parte lessicale (correttezza di scrittura e riconoscibilità del comando rispetto al lessico previsto) sia per quanto concerne la possibilità di fare quella particolare morsa. Sarete dunque informati da appropriati messaggi nel caso di una vostra erronea indicazione; verrete invece spostati nella direzione richiesta, se l'input ha superato tutti i meccanismi di controllo. Una variante di tale procedimento riguarda comandi costituiti da più parole, vale a dire istruzioni rivolte a realizzare una data azione, quali "Prendo Martello" o "Metto Stella Buco". Queste istruzioni complesse devono necessariamente contenere un verbo e un solo oggetto; già la richiesta di mettere la stella nel buco è

Descrizione del programma

Linee	Commento
10 - 30	inizializzazione
40 - 80	gestione dei premi nelle stanze
90 - 190	accoglimento dei comandi inseriti da tastiera
200 - 330	verifica dei comandi inseriti da tastiera, conseguente salto a stanza successiva o ad esecuzione di una data azione
1000 - 1007	presentazione
1010 - 2310	stanze del castello
3000 - 3620	verifica della direzione prescelta
3700 - 3870	routine per la raccolta o l'abbandono dei diversi premi
3900 - 5799	routine per l'esecuzione di azioni particolari (spostamenti di suppellettili nelle stanze, assemblamento di oggetti per costruirne un altro, ecc.).
5800 - 5860	memorizzazione su nastro della situazione ad un dato momento, per interrompere la partita
5900 - 5920	caricamento da nastro della situazione di una partita precedentemente interrotta (e salvata)
6000 - 6150	routine associata ai comandi LISTA e INVENTARIO
6200 - 6290	routine associata al comando AIUTO, per ottenere suggerimenti
6300 - 6710	altre azioni particolari
7000 - 7020	verifica dei premi a vostra disposizione
9000 - 9050	dati per la gestione del gioco
9930 - 9940	azzeramento del contatore del tempo
9950 - 9990	svariati tipi di morte

Descrizione delle variabili

N,S,E,W	spostamenti nelle quattro direzioni cardinali
U,D	spostamenti verso l'alto e il basso
EN,L	ingresso ed uscita dalle stanze
R()	puntatore; invia il programma alla successiva stanza
O()	disposizione degli oggetti nel palazzo
C()	premi raccolti dal giocatore
D()	controllo sull'apertura delle porte
LIFE, FN A()	tempo a disposizione
LLIFE	controllo immortalità

del tutto eccezionale e vi sarà suggerita dal programma stesso. Anche in questo caso il programma verifica la possibilità di eseguire il comando secondo le modalità prima esposte. Vediamo ora di comprendere come il programma può riconoscere la planimetria del castello.

Questa è concretamente gestita dalle linee 1010 - 2125. Ogni locale in cui potete entrare è, sotto il profilo operativo, gestito da una coppia di linee: una linea contenente una istruzione PRINT, atta a descrivere la configurazione della stanza, e una linea contenente una o più istruzioni LET (talvolta anche una GO SUB), che determina gli spostamenti consentiti da quella particolare posizione della casa. Ad esempio, la linea 1010 vi informa circa l'aspetto della stanza in cui vi trovate, la linea 1015 assegna alle variabili W (WEST), E (East) e N (North) tre distinti valori; scegliendo una di queste tre direzioni si assegna il valore della corrispondente variabili ad un puntatore R(1) e che realizza il salto nella linea di programma (cioè ad un'altra stanza) per mezzo del seguente algoritmo:

GO SUB (R(1) * 10) + 1000

Se nel nostro esempio scegliamo di spo-

starci ad Est, direzione a cui corrisponde la variabile E=2, accederemo alla stanza associata alla linea:

$$(2 * 10) + 1000 = 1020$$

Quando invece realizziamo un'azione particolare, come raccogliere un oggetto, l'algoritmo che viene considerato è:

$$GO TO (F * 100) + 3000$$

dove F si riferisce alla particolare operazione richiesta. Attraverso questa procedura si accede alla sezione del listato destinata alla gestione delle situazioni particolari; terminata l'azione si ritorna alle routine di controllo, pronti all'inserimento di un nuovo ordine.

Il gioco consente di contare su un lessico di oltre 60 termini, fra verbi e oggetti; di particolare interesse le opzioni "LISTA" e "INVENTARIO", per sapere in ogni istante quali oggetti sono in nostro possesso; "AIUTO", per ricevere talvolta opportuni suggerimenti; infine "SALVA" e "CARICA", opzioni grazie alle quali possiamo interrompere una partita per riprenderla successivamente, salvando e caricando da nastro gli array che controllano la nostra posizione, i premi raccolti, ecc..

Prima di iniziare a muovere i primi passi all'interno della "Casa del Destino" consentiteci di sottolineare che il programma è stato accuratamente verificato: la soluzione di questa avventura è realmente molto difficile, ma non impossibile. Nel caso non riusciate a guadagnare l'uscita nel giro di qualche partita non attribuite precipitosamente la colpa dell'insuccesso al listato, ma aguzzate l'ingegno e, soprattutto, liberate la vostra fantasia: vedrete che la vostra tenacia sarà adeguatamente ricompensata dal lieto fine.

inserire: DESCRIZIONE DEL PROGRAMMA

Vocabolario a disposizione

Comandi

N
E
S
O
Nord
Est
Sud
Ovest
Alto
Basso
A
B
Prendo
Lascio
Faccio
Blocco
Scaldo
Allaccio
Lego
Metto
Aprò
Leggo
Uso
Mangio
Getto
Sblocco
Bevo
Salgo

Entro
Dentro
Fuori
Esco
Muovo
Salva
Carica
Guardo
Lista
Inventario
Aiuto

Notate che si gioca in prima persona, tranne nel caso dei comandi SALVA, CARICA e LISTA. Fin d'ora vi suggeriamo che il verbo "allaccio" può essere utilizzato proficuamente in un solo ambiente; che non sempre si possono usare i termini ENTRO e ESCO; che la richiesta di AIUTO spesso non fornisce suggerimenti utili, ma è quasi indispensabile per raggiungere la soluzione; che SALGO diviene importantissimo vicino alla fine del gioco; infine che GUARDO consente di ricontrollare la vostra posizione nella stanza, nel caso in cui sia sparita la didascalìa.

Oggetti

Fune
Scaletta
Stella
Liquido verde
Libro
Scatola di sigari
Chiodi

Martello
Legno
Medaglione d'oro
Lenzuolo
Lingotto d'oro
Stampo
Scopa
Rubino
Gancio
Chiave blu
Chiave rossa
Lente
Sapone

Giocando vi accorgete che alcuni oggetti non si trovano mai in giro! Pensate approfonditamente a questo curioso aspetto: ne trarrete vantaggio.

Altri termini che talvolta il programma riconosce

Buco
Scaffale
Tappeto
Quadri
Porta
Frutta
Vino
Apriti Sesamo

Questi termini non rappresentano oggetti che potete raccogliere, ma la realizzazione di un'azione su di essi è indispensabile per vincere la partita.

```
10 LET LLIFE=1: LET LIFE=2000:
   BORDER 6: PAPER 7: INK 0:
   GO SUB 1000: POKE 23658,8:
   DIM R(2): DIM O(30): DIM C(5):
   DIM D(10,2)
20 RESTORE 9050: LET R(1)=1: FOR F=
  1 TO 30: READ O(F): NEXT F
30 LET L=0: LET EN=0: LET N=0:
   LET E=0: LET S=0: LET W=0:
   LET U=0: LET D=0: GO SUB (R(1)*
  10)+1000: LET H=0
40 FOR F=1 TO 30: IF O(F)=R(1)
   THEN GO TO 60
50 NEXT F: GO TO 90
60 IF H=0 THEN PRINT INK 1;"INOL
  TRE VEDI:"
```

```
70 LET H=1: RESTORE 9040
80 FOR G=1 TO F: READ C$: NEXT G:
  PRINT C$: GO TO 50
90 LET T=1: PRINT AT 18,0;"]-":
  LET A$=""
94 IF LLIFE=0 THEN GO TO 100
95 IF FN A(>)>LIFE THEN GO TO 9950
100 FOR G=1 TO 100: LET B$=INKEY$:
  IF B$<>" " THEN GO TO 110
105 NEXT G: GO TO 94
110 BEEP .01,10: IF CODE B$=12
  THEN GO TO 160
120 IF CODE B$=13 AND A$<>" "
  THEN GO TO 200
130 IF CODE B$=13 THEN GO TO 100
135 IF LEN A$>29 THEN GO TO 100
140 LET A$=A$+B$: PRINT AT 18,T;B$;"
  -"
145 IF INKEY$<>" " THEN GO TO 145
```

```

150 LET T=T+1: GO TO 100
160 IF A$="" THEN GO TO 100
170 LET T=T-1: LET A$=A$( TO T-1)
180 PRINT AT 18,T;"- "
185 IF INKEY$<>"" THEN GO TO 180
190 GO TO 100
200 IF R(1)=126 AND R(2)=3 AND A$="S
ALGO" THEN LET F=34: GO TO 330
201 IF LEN A$<6 THEN GO TO 205
202 IF A$(1 TO 6)="APRITI" THEN
LET A$="APRO"+A$(7 TO LEN A$)
205 LET D$="": LET FF=1: CLS :
PRINT ">"; INK 1;A$: IF A$(
LEN A$)<>"." THEN LET A$=A$+"
."
210 RESTORE 9000: FOR F=FF TO LEN A$

220 IF A$(F)=" " OR A$(F)="." THEN
GO TO 240
230 NEXT F: GO TO 290
240 IF A$(FF TO F-1)="" THEN
GO TO 230
250 LET B$=A$(FF TO F-1): FOR G=1
TO 40: READ C$
260 IF C$=B$ THEN GO TO 280
270 NEXT G: LET FF=F+1: GO TO 210
280 RESTORE 9010: FOR H=1 TO G:
READ C: NEXT H: LET D$=D$+
STR$ C: LET FF=F+1: GO TO 210
290 RESTORE 9020: FOR F=1 TO 37:
READ C$
300 IF C$=D$ THEN GO TO 330
310 NEXT F
320 PRINT "NON CONOSCO UN SIMILE CO
MANDO.": GO TO 90
330 GO TO (F*100)+3000
1000 CLS : PRINT INK 2;AT 1,2;"PER A
MORE DI UNA RAGAZZA..."
1005 PRINT "TI SEI PERDUTAMENTE INNA
MORATO DI UNA RAGAZZA RICCHISSI
MA DI NOME ELENA, E POICHE'
TU SEI POVERO, TI SEI IMPOSTO
UNA DIFFICILE PROVA PER DIMO
STRARLE IL TUO AMORE (E PER DIMO
STRARE AL DI LEI RICCO GENITORE
IL TUO COMPLETO DISINTERESSE):"

1006 PRINT "DEVI RIUSCIRE A SCAPPARE
DAL 'PALAZZO DEL DESTINO' DO
PO AVER TROVATO UN RUBINO E UN L
INGOTTO D'ORO.....BUONA FORTUNA!
!!"
1007 PRINT AT 21,0;"PREMI UN TASTO
PER CONTINUARE": PAUSE 2000:
GO SUB 9930: CLS : RETURN
1010 PRINT "SEI DI FRONTE AL PORTONE
. E' MOLTO GRANDE E SEMBRA PE
SANTE. 4 GROSSI CATENACCI MANTE
NGONO LAPORTA CHIUSA. DI FRONTE,
A NORD ELONTANO DALLA PORTA,C'E'
UNA RAMPA DI SCALE. 2 CORRID
OI VANNOAD EST E AD OVEST."
1015 LET W=21: LET E=2: LET N=118:
RETURN

```

```

1020 PRINT "SEI IN UN LUNGO CORRIDOI
O CHE VADA EST A OVEST.UNA ROBUS
TA PORTASI APRE NEL MURO A NORD.
"
1024 IF D(1,2)=1 THEN GO SUB 3000
1025 LET E=4: LET W=1: IF D(1,2)=1
THEN LET EN=119
1026 RETURN
1030 PRINT "SEI QUASI AL TERMINE NOR
D DI UN CORRIDOIO. UNA ROBUSTA P
ORTA E' SOCCHIUSA. LE USCITE SON
O A NORDE A SUD."
1034 IF D(2,2)=1 THEN GO SUB 3000
1035 LET N=19: LET S=119: RETURN
1040 PRINT "SEI IN UN LUNGO CORRIDOI
O CHE VADA EST A OVEST. UNA POR
TA SI TROVA NEL MURO A SUD."
1044 IF D(2,2)=1 THEN GO SUB 3000
1045 LET E=6: LET W=2: IF D(2,2)=1
THEN LET EN=5
1046 RETURN
1050 PRINT "SEI IN UN VESTIBOLO, A F
IANCO ADUN BAGNO. UNA PORTA E' L
'USCITA."
1055 LET L=4: RETURN
1060 PRINT "SEI ALL'INCROCIO FRA I C
ORRIDOI.PUOI ANDARE A OVEST,NORD
O SUD."
1065 LET W=4: LET N=16: LET S=7:
RETURN
1070 PRINT "SEI AD UN BIVIO NEL CORR
IDOIO. PUOI ANDARE A NORD O A E
ST."
1073 LET E=122
1074 IF D(3,2)=1 THEN LET EN=8
1075 LET N=6: RETURN
1080 PRINT "TI TROVI NELLA STANZA
DEGLI OSPITI, DOVE SI SONO
SVOLTI INFINITI PARTIES. LE USC
ITE SONOA NORD E A OVEST."
1085 LET W=7: LET N=9: RETURN
1090 PRINT "TI TROVI NELLA SALA DA B
ALLO. LE USCITE SONO A NORD E
A SUD."
1095 LET N=10: LET S=8: RETURN
1100 PRINT "SEI IN UNA GALLERIA. QUA
DRI SONOAPPESI ALLE PARETI. LA G
ALLERIA E' MOLTO LUNGA E STRETTA
. USCITEA NORD,OVEST E SUD."
1105 LET N=12: LET W=11: LET S=9:
RETURN
1110 PRINT "SEI IN UN CORTO CORRIDOI
O. SI VEDONO DUE PORTE ALLE PA
RETI. USCITA A EST."
1113 IF D(5,2)=1 THEN PRINT "LA POR
TA A NORD E' APERTA."
1114 IF D(4,2)=1 THEN PRINT "LA POR
TA A SUD E' APERTA."
1115 LET E=10: IF D(5,2)=1 THEN
LET N=14
1116 IF D(4,2)=1 THEN LET S=13
1117 RETURN
1120 PRINT "SEI IN UNA STANZA DESTIN
ATA ALLOSVAGO E VARI GIOCHI SONO

```

```

SPARSI DAPPERTUTTO.LE USCITE SO
NO A SUDE A OVEST."
1125 LET W=15: LET S=10: RETURN
1130 PRINT "'SEI IN UNO STUDIO.UNA SC
RIVANIA E' AL CENTRO DELLA STANZ
A CON QUALCHE SEDIA INTORNO."
1135 LET L=11: LET N=11: RETURN
1140 PRINT "'SEI NELLA LIBRERIA. UN N
OTEVOLE NUMERO DI LIBRI E' SU SC
AFFALI CHE CORRONO LUNGO LA STA
NZA. MOLTI VOLUMI SONO AMMU
CCHIATI SOPRA GLI STESSI LIBRI."

1145 LET L=11: LET S=11: RETURN
1150 PRINT "'SEI IN UN PICCOLO CORRID
OIO. HAIDI FRONTE A TE UNA PORTA
,MENTRE IL CORRIDOIO VA A SUD E
A EST."

1154 LET E=12: GO SUB 3000: LET S=120
: LET EN=18: RETURN
1160 PRINT "'SEI AL TERMINE DEL CORRI
DOIO E C'E' UNA PORTA ALLA PARE
TE. USCITA A SUD."
1165 LET LIFE=2000: GO SUB 3000:
LET EN=17: LET S=6: RETURN
1170 PRINT "'SEI IN UN SALONE,MA E' V
UOTO. TISUGGERISCO DI ANDARTEN
E IN FRETTA."
1175 LET LIFE=FN A()+10: LET L=16:
RETURN
1180 PRINT "'SEI IN UNA NON USATA CAM
ERA DA GIOCHI DEI BAMBINI."
1185 LET L=15: RETURN
1190 PRINT "'SEI AL TERMINE DEL CORRI
DOIO.UNAPORTA SI TROVA NELLA PA
RETE AD OVEST. USCITA A SUD."
1195 LET S=3: GO SUB 3000: LET EN=20:
RETURN
1200 PRINT "'SEI IN UN SALOTTO. LA ST
ANZA E' GRANDE MA SEMBRA MOLTO I
NTIMA. QUALCOSA NON TI CONVINCEN
."
1203 IF D(10,2)=1 THEN LET D=76
1204 IF D(10,2)=1 THEN LET EN=76
1205 LET L=19: RETURN
1210 PRINT "'SEI IN UN CORRIDOIO ORIE
NTATO A EST/OVEST.UNA PORTA SI T
ROVA NELMURO A SUD."
1214 IF D(6,2)=1 THEN GO SUB 3000
1215 LET E=1: LET W=23
1216 IF D(6,2)=1 THEN LET EN=22
1217 RETURN
1220 PRINT "'CAMMINI NELLA STANZA E L
A PORTA SI RICHIUDE ALLE TUE SPA
LLE.NOTIIN GIRO GLI SCHELETRI DI
ALTRI CHE HANNO TENTATO LA PRO
VA."
1221 PAUSE 150: PRINT "'TENTI DI APRI
RE LA PORTA MA NON SI MUOVE. SEM
BRA CHE IL MEDESIMODESTINO TI AT
TENDA."
1225 GO TO 9960
1230 PRINT "'SEI IN UN CORRIDOIO ORIE
NTATO A EST/OVEST.UNA PORTA SI T

```

LOAD

For love

```

ROVA NELMURO A SUD."
1235 GO SUB 3000: LET E=21: LET W=25:
LET EN=24: RETURN
1240 PRINT "'SEI NELLA CUCINA. UNA CA
SSERUOLAE' SOPRA UNA STUFA ANCOR
A CALDA."
1245 LET L=23: RETURN
1250 PRINT "'SEI ALL'INCROCIO FRA I C
ORRIDOI.PUOI ANDARE A EST,NORD O
SUD."
1255 LET S=26: LET N=27: LET E=23:
RETURN
1260 PRINT "'SEI IN CIMA A UNA RAMP
A DI SCALEED UN CORRIDOIO VA A NOR
D."
1265 LET D=36: LET N=25: RETURN
1270 PRINT "'SEI IN UN CORTO CORRI
DOIO ORIENTATO A NORD/SUD. UN
A PORTA SI TROVA NEL MURO A EST.
"
1275 GO SUB 3000: LET EN=28: LET N=29
: LET S=25: RETURN
1280 PRINT "'SEI NELLA SALA DA PRANZO
.NELLA STANZA C'E SOLO UN LUNGO
TAVOLO MENTRE LA PARETI SONO SP
OGLIE."
1285 LET L=27: RETURN
1290 PRINT "'SEI IN UN CORTO CORRI
DOIO ORIENTATO A NORD/SUD. UN
A PORTASI TROVA NEL MURO A EST.
"
1295 LET N=31: LET S=27: GO SUB 3000:
LET EN=30: RETURN
1300 PRINT "'SEI IN UNA STANZA DA BAG
NO CON TOILETTE. SEMBRA MOLTO S
PORCA."
1305 LET L=29: RETURN
1310 PRINT "'SEI AD UN BIVIO NEL CORR
IDOIO. PUOI ANDARE A SUD O A ES
T."
1315 LET S=29: LET E=32: RETURN
1320 PRINT "'SEI AD UN BIVIO NEL CORR
IDOIO. PUOI ANDARE A SUD O A OV
EST."
1325 LET S=33: LET W=31: RETURN
1330 PRINT "'SEI IN FONDO AD UN COR
RIDOIO MOLTO CORTO.UNA PORTA SI
TROVA NEL MURO OVEST. USCITA A
NORD."
1335 LET N=32: GO SUB 3000: LET EN=34
: RETURN
1340 PRINT "'SEI IN UNA STANZA PER IL
CUCITO.IL LAVORO SEMBRA INTERRO
TTO DA POCO."
1345 LET L=33: RETURN
1350 PRINT "'SEI NEL RIPOSTIGLIO DELL

```

```

E SCOPE.USCITA IN ALTO."
1355 LET U=118: RETURN
1360 PRINT "SEI AGLI ULTIMI GRADINI
DI UNA SCALA DELLE CANTINE. QUE
STO LOCALE E' PIENO DI RIFIU
TI. UNA PORTA RAPPRESENTA L'USCI
TA."
1365 LET U=26: LET L=37: GO SUB 3000:
RETURN
1370 PRINT "SEI NEL CORRIDOIO DELLA
CANTINA.IL CORRIDOIO VA VERSO NO
RD E DUEPORTE SONO A EST E OVEST
."
1371 IF D(7,2)=0 THEN PRINT "SOLO LA
PORTA A OVEST E' APERTA."
1372 IF D(7,2)=1 THEN PRINT "LE DUE
PORTE SONO APERTE."
1373 IF D(7,2)=1 THEN LET E=38
1375 LET N=39: LET W=36: RETURN
1380 PRINT "SEI IN UN LABORATORIO.GL
I ARNESISONO APPESI ALLE PARETI.
"
1385 LET L=37: RETURN
1390 PRINT "SEI AD UN INCROCIO."
1395 LET N=45: LET E=40: LET W=42:
LET S=37: RETURN
1400 PRINT "SEI IN FONDO A UN CORRID
OIO. UNAPORTA SI TROVA NELLA PA
RETE A NORD. USCITA A OVEST."
1405 LET W=39: GO SUB 3000: LET EN=41
: RETURN
1410 PRINT "SEI IN UNA PICCOLA STANZ
A DOVE C'E' UNA LUCE MOLTO FIOC
A, MA RIESCI A VEDERE ALCUNE S
CRITTE SUL MURO."
1415 LET L=40: RETURN
1420 PRINT "SEI IN FONDO A UN CORRID
OIO. DUEPORTE SONO APERTE ED UNO
STRANO ODORE PROVIENE DA UNA DI
ESSE. LE PORTE SONO A NORD E
A SUD."
1425 LET S=43: LET N=44: LET E=39:
RETURN
1430 PRINT "LA STANZA E' COLMA DI FR
UTTA."
1435 LET L=42: RETURN
1440 PRINT "LA STANZA E' COLMA DI FR
UTTA."
1445 LET L=42: RETURN
1450 PRINT "SEI IN UN CORRIDOIO ORIE
NTATO A NORD/SUD.UNA PORTA SI T
ROVA NELMURO A OVEST."
1455 LET N=47: LET S=39: GO SUB 3000:
LET EN=46: RETURN
1460 PRINT "SEI IN UNA STANZA CON
TUTTI I MOBILI DISTRUTTI."
1465 LET L=45: RETURN
1470 PRINT "SEI AL TERMINE NORD D
I UN CORRIDOIO. LA PORTA E' S
ITUATA NEL MURO AD OVEST."
1475 LET S=45: GO SUB 3000: LET EN=49
: RETURN
1480 PRINT "SEI GIUNTO AD UN PIANERO
TTOLO. SCALINI VANNO A SUD E A

```

```

NORD."
1485 LET LIFE=2000: LET S=53: LET N=5
4: RETURN
1490 PRINT "SEI NELLA CANTINA DEI
VINI. LE BOTTIGLIE SONO SU S
CAFFALI.BICCHIERI DI VINO SONO
SU DI UN VASSOIO"
1495 LET L=47: RETURN
1500 PRINT "SEI IN UNA CELLA GRANDE
COME UNASTANZA. LE PARETI SONO
NUDE. LEUSCITE SONO A EST E A NO
RD."
1505 LET E=51: LET N=53: RETURN
1510 PRINT "SEI IN UNA STANZA CIRCOL
ARE. UN GRANDE BUCO E' SOPRA DI
TE."
1515 LET LIFE=2000: LET N=52: LET W=5
0: RETURN
1520 PRINT "SEI IN UNA STANZA POLVER
OSA. TI SENTI MANCARE IL RESPIRO
, QUINDIESCI IN FRETTA"
1525 LET LIFE=FN A()+10: LET S=51:
LET L=51: RETURN
1530 PRINT "SCALINI VANNO VERSO L'AL
TO E LA PORTA E' A SUD"
1535 LET S=50: LET U=48: RETURN
1540 PRINT "SEI IN UNA STANZA IN CUI
ALEGGIAUN BRUTTO ODORE.TI SENTI
COME SEDEGLI OCCHI TI OSSERVA
SSERO. DECIDITI IN FRETTA E DIR
IGITI A EST O SUD."
1545 LET LIFE=FN A()+10: LET E=55:
LET S=48: RETURN
1550 PRINT "TI TROVI NELLA CAMERA OS
CURA.UNOSCAFFALE SEMBRA FUORI PO
STO."
1555 LET LIFE=2000: LET E=54: LET L=5
4: RETURN
1560 PRINT "IL MURO SI SPOSTA."
1565 LET R(1)=119: GO TO 2190
1570 PRINT "LO SCAFFALE SI MUOVE."
1575 LET R(1)=55: GO TO 1550
1580 PRINT "TI TROVI IN UNA STANZA S
EGRETA DIETRO LA LIBRERIA."
1585 LET L=57: RETURN
1590 PRINT "SEI IN UNA CAMERA DA LET
TO. LE PARETI SONO VIVACEMENTE
DECORATEE TUTTO SEMBRA CONFORTE
VOLE."
1595 LET L=60: RETURN
1600 PRINT "SEI IN UN CORRIDOIO.A SU
D C'E'UNVICOLO CIECO MA VI SONO
PORTE AILATI EST E OVEST.USCITA
A NORD."
1605 LET N=63: LET E=61: LET W=59:
RETURN
1610 PRINT "SEI IN UNA CAMERA DA LET
TO PER GLI OSPITI PIU' IMPORTAN
TI. IL LETTO E IL BALDACCHINO S
EMBRANO MOLTO ELEGANTI. UNA POR
TA E' L'USCITA,UN'ALTRA CONDUC
E A EST."
1615 LET E=62: LET L=60: LET W=60:
RETURN

```



```

1620 PRINT "SEI NEL BAGNO RISERVAT
O AGLI OSPITI. LA PORTA E' L'US
CITA."
1625 LET W=61: LET L=61: RETURN
1630 PRINT "SEI AD UNA SVOLTA NEL CU
NICOLA. UNA PORTA E' NEL MURO OV
EST. LE USCITE SONO A EST E A SU
D."
1635 GO SUB 3000: LET EN=64: LET S=60
: LET E=67: RETURN
1640 PRINT "SEI IN UNA STANZA DA LET
TO. UNA BRANDA E' CONTRO UNA PAR
ETE."
1645 LET L=63: RETURN
1670 PRINT "SEI IN UN PIANEROTTOLO O
RIENTATO A EST/OVEST CON UNA PORT
A APERTAVERSO SUD."
1675 LET W=63: LET E=123: LET EN=68:
LET S=68: RETURN
1680 PRINT "SEI IN UN SOGGIORNO RISE
RVATO AGLI OSPITI. SUL PAVIMEN
TO CALDIE MORBIDI TAPPETI."
1685 LET L=67: LET N=67: RETURN
1690 PRINT "SEI IN UN PIANEROTTOLO O
RIENTATO A EST/OVEST CON UNA PORT
A APERTAVERSO SUD."
1695 LET W=123: LET E=71: GO SUB 3000
: LET EN=70: RETURN
1700 PRINT "SEI IN UNA CAMERA DA LET
TO DALLE NUDE PARETI. E' UMIDA E
FREDDA."
1705 LET L=69: RETURN
1710 PRINT "TI TROVI AD UN INCROCIO
SUL PIANEROTTOLO. USCITE A N
ORD, EST, SUD E OVEST."
1715 LET N=72: LET S=80: LET W=69:
LET E=77: RETURN
1720 PRINT "SEI AL DI FUORI DI UNA P
ORTA. USCITA VERSO SUD."
1725 LET S=71: GO SUB 3000: LET EN=73
: RETURN
1730 PRINT "SEI IN UNA CAMERA DA LET
TO. UN GUARDAROBA E' IN UN ANGO
LO."
1734 IF D(8,2)=0 THEN LET EN=74
1735 LET L=72: RETURN
1740 PRINT "SEI DENTRO IL GUARDAROBA
. E' VUOTO. USCITE A EST E OV
EST."
1745 LET E=73: LET W=75: RETURN
1750 PRINT "SEI IN UNA PICCOLA ST
ANZA NASCOSTA DAL GUARDAROBA.
"
1755 LET L=73: RETURN
1760 PRINT "SEI IN UN BUCO DEL PAVIM
ENTO."
1764 LET D(10,1)=0
1765 LET U=110: RETURN
1770 PRINT "SEI AL DI FUORI DI UNA P
ORTA. USCITA VERSO OVEST."
1775 LET W=71: GO SUB 3000: LET EN=78
: RETURN
1780 PRINT "SEI IN UNA STANZA DA LET
TO. LA CAMERA E' SPOGLIA E SENZ

```

```

A VITA."
1785 LET L=77: RETURN
1800 PRINT "SEI IN FONDO ALLE SCA
LE. LE USCITE SONO IN ALTO E A
NORD."
1805 LET U=99: LET N=71: RETURN
1810 PRINT "SEI NELLA CAMERA DI UNA
RAGAZZA. IL LETTO E' IN ORDINE E L
A CAMERA E' MOLTO PULITA."
1815 LET L=88: RETURN
1820 PRINT "SEI NELLA CAMERA DI UNA
RAGAZZA IN CUI TUTTO SEMBRA TRAS
ANDATO."
1825 LET L=87: RETURN
1830 PRINT "SEI NELLA STANZA DEI DOM
ESTICI. UNO SPECCHIO E' APPESO A
L MURO."
1840 LET L=86: RETURN
1850 PRINT "UNA DEVIAZIONE NEL PIANE
ROTTOLO. LE USCITE SONO A EST E A
SUD."
1855 LET E=102: LET S=86: RETURN
1860 PRINT "SEI IN UN PIANEROTTOLO O
RIENTATO A NORD/SUD. UNA PORTA
E' ALLA PARETE OVEST."
1865 LET EN=83: LET N=85: LET S=87:
GO SUB 3000: RETURN
1870 PRINT "SEI IN UN PIANEROTTOLO O
RIENTATO A NORD/SUD. PORTE A EST
E OVEST. ENTRAMBE SONO APERTE."
1875 LET E=91: LET W=82: LET N=86:
LET S=88: RETURN
1880 PRINT "UNA DEVIAZIONE NEL PIANE
ROTTOLO. UNA PORTA E' ALLA PARETE
OVEST. LE USCITE SONO A EST E A
NORD."
1885 LET N=87: LET E=89: GO SUB 3000:
LET EN=81: RETURN
1890 PRINT "SEI AL DI FUORI DI UNA P
ORTA. USCITA VERSO OVEST."
1895 LET W=88: GO SUB 3000: LET EN=94
: RETURN
1910 PRINT "SEI NEGLI ALLOGGI PER LA
NOTTE DEI CUOCHI."
1915 LET L=87: RETURN
1920 PRINT "SEI NELLA LAVANDERIA MA
NON VEDIBIANCHERIA IN GIRO. UNA
VASCA E' SUL PAVIMENTO MA E' TUTT
A ROTTA."
1925: LET L=99: RETURN
1940 PRINT "SEI NELLA STANZA DEI SER
VI. E' ESTREMAMENTE ORDINATA E
NULLA SEMBRA FUORI POSTO."
1945 LET L=89: RETURN
1970 PRINT "SEI NELLA SALA DA MUSICA

```

LOAD

For love

MA NON C'E' NULLA A PARTE UNO
 SPESSO STRATO DI POLVERE."
 1975 LET L=98: LET N=98: RETURN
 1980 PRINT "'SEI NELLA SALA DA MUSICA
 . SU UN LEGGIO VI E' UNO SPARTIT
 O. UNA PORTA APERTA E' A SUD"
 1985 LET EN=97: LET W=99: LET S=97:
 LET L=99: RETURN
 1990 PRINT "'TI TROVI IN CIMA ALLE SC
 ALE. UN PIANEROTTOLO VA A NORD E
 PORTE APERTE SONO A EST E A OV
 EST."
 1995 LET D=80: LET W=92: LET E=98:
 LET N=100: RETURN
 2000 PRINT "'IL PIANEROTTOLO VA A NOP
 D, SUD EEST,UNA PORTA E' NEL MUR
 O OVEST."
 2005 LET N=102: LET S=99: LET E=104:
 GO SUB 3000: LET EN=101:
 RETURN
 2010 PRINT "'SEI NELL'AULA IN CUI IL
 TUTORE SUOLE TENERE LE LEZIONI.
 SULLA LAVAGNA C'E' UNA ADDIZIO
 NE. LA OPERAZIONE E' 13+15, E T
 UTTO VIENE POI MOLTIPLICATO P
 ER 100."
 2015 LET L=100: RETURN
 2020 PRINT "'UNA DEVIAZIONE NEL PIANE
 ROTTOLO.PUOI ANDARE A OVEST O SU
 D E UNA PORTA E' A EST."
 2025 LET W=85: LET S=100: GO SUB 3000
 : LET EN=103: RETURN
 2030 PRINT "'SEI NELLA CAMERA DEL TUT
 ORE. UNAPORTA E' APERTA VERSO ES
 T."
 2035 LET L=102: LET EN=106: LET E=106
 : RETURN
 2040 PRINT "'SEI DAVANTI A TRE PORTE
 APERTE. L'USCITA E' A OVEST,MENT
 RE LE PORTE SONO A NORD,SUD E
 EST."
 2045 LET W=100: LET E=111: LET N=108:
 LET S=112: RETURN
 2060 PRINT "'SEI NEL BAGNO PERSONAL
 E DEL TUTORE. L'USCITA E' UNA
 PORTA."
 2065 LET L=103: RETURN
 2080 PRINT "'SEI IN CAMERA DELLA BAMB
 INAIA. PORTE VERSO EST E SUD. U
 NA CULLADONDOLA AVANTI E INDIETR
 O."
 2085 GO SUB 3000: LET E=110: LET S=10
 4: LET L=104: LET EN=110:
 RETURN
 2100 PRINT "'SEI NELLA CAMERA DA LETT
 O DELLA BAMBINAIA, STANZA PICCOL
 A MA PIACEVOLE. UNO STRANO TA
 PPEO E'SUL PAVIMENTO."
 2104 IF D(10,1)=1 THEN PRINT "UN BUC
 O NEL PAVIMENTO! SCENDI !":
 LET D=76
 2105 LET L=108: LET W=108: RETURN
 2110 PRINT "'SEI NELLA STANZA DEI GIO
 CHI DEI BAMBINI. ALCUNI GIOCATTO

LI SONO SPARSI PER TERRA."
 2115 LET L=104: LET W=104: RETURN
 2120 PRINT "'SEI IN UNA PICCOLA STANZ
 A PIENA DI DISEGNI DEI BAMBINI S
 UI MURI."
 2125 LET L=104: LET N=104: RETURN
 2130 PRINT "'QUATTRO DIREZIONI. NORD,
 EST,SUD E OVEST."
 2135 LET N=114: LET S=116: LET W=117:
 LET E=115: RETURN
 2140 PRINT INK 3;"CAMMINI NELLA STA
 NZA E LA PORTA SI CHIUDE. LE PA
 RETI SEMBRANO RIDERE E POI PIAN
 GERE. SPECCHI RIFLETTONO LA TUA
 IMMAGINE E LA DEFORMANO. LA LUC
 E TI ACCECA E GRIDA TI ASSORDAN
 O. LA PORTA SI E' RICHIUSA SUL T
 UO DESTINO."
 2145 GO TO 9980
 2150 PRINT INK 2;"CORRI A OVEST!":
 LET LIFE=FN A()+5: LET LLIFE=1:
 LET W=113: GO TO 90
 2160 PRINT "'UN LUNGO SCIVOLO DI FRON
 TE A TE VA IN GIU' VERSO IL TERR
 ENO. FUORI UNA FOLLA SI E' RA
 DUNATA E,AL TUO APPARIRE,TUTTI
 INIZIANOAD APPLAUDIRE! MA HAI VI
 NTO ?"
 2165 PRINT "'SCIVOLI VERSO IL TERRENO
 . DEVI ORA CONSEGNARE IL LINGOT
 TO D'OROE IL RUBINO AL PADRE DI
 ELENA. TUTTI SONO IN ATTESA."
 2166 PAUSE 600: LET D\$="RUBINO":
 GO SUB 7000: IF H=6 THEN
 PRINT "'OH CARO! NON HAI IL RUB
 INO. COSAFARAI ADESSO ? CERTO NO
 N POTRAI SPOSARE ELENA. HAI FAL
 LITO LA TUA PROVA.": GO TO 9980
 2167 LET D\$="LINGOTTO D'ORO":
 GO SUB 7000: IF H=6 THEN
 PRINT "'OH CARO! HAI FALLITO LA
 PROVA POICHE' NON HAI IL LING
 OTTO. BEH, PUOI SEMPRE RIPROV
 ARCI.": GO TO 9980
 2168 CLS : PRINT INK 1;"SEI RIUSCITO
 NEL TUO INTENTO. ELENA CORRE
 VERSO DI TE CON LE BRACCIA APER
 TE PER ABBRACCIARTI.SUO PADRE E'
 RAGGIANTE. LUNGA VITA A
 TUTTI E DUE."
 2169 BEEP .9,-5: BEEP .5,0: PAUSE 1:
 BEEP .15,0: BEEP 1,0: PAUSE 2:
 BEEP .9,-5: BEEP .5,2: PAUSE 1:
 BEEP .15,-1: BEEP 1,0: PAUSE 2:
 BEEP .9,-5: BEEP .5,0: PAUSE 1:
 BEEP .15,4: BEEP .5,7: PAUSE 1:
 BEEP .15,4: BEEP .9,0: PAUSE 2:
 BEEP .9,-5: BEEP .5,2: PAUSE 1:
 BEEP .15,-1: BEEP 1,0: PAUSE 10
 : GO TO 2169
 2170 PRINT "'CASCHI IN UNA BOTOLA
 NEL PAVIMENTO. NE AFFERRI IL
 BORDO E TI TIRI SU.

```

                MA ECCO APPARIRE UN PIED
ONE. E' ENORME E MINACCIOSO E
                TI SCHIACCIA LE DITA."
2175 PRINT "PRECIPITI VERSO LA MORTE
." : GO TO 9980
2180 PRINT "SEI AI PIEDI DI UNA SCAL
A. NELLARINGHIERA C'E' UNO STRAN
O BUCO AFORMA DI STELLA. LE USCI
TE SONOA SUD E IN ALTO."

2185 LET U=123: LET S=1: RETURN
2190 PRINT "'TI TROVI IN UN CORRIDOIO
LUNGO ESTRETTO CHE PORTA A NORD
E SUD. SULLA PARETE OVEST CI SO
NO DEI QUADRI, MENTRE IL MURO E
ST E' SPOGLIO. TUTTO TI SEMBRA
STRANO."
2195 LET N=3: LET S=2: RETURN
2200 PRINT "SEI IN UNA SERRA.ALCUNE
SCULTUREDECORANO IL LOCALE."
2205 LET L=15: LET N=15: RETURN
2220 PRINT "SEI IN UNA STANZA E UNA
PORTA E'DALL'ALTRO LATO. POTREST
I RAGGIUNGERLA SALENDO SU
DI UN NASTRO TRASPORTATORE,MA
VA NELLADIREZIONE SBAGLIATA.IL P
ULSANTE PER FERMARE IL NASTRO E'
SULLA PARETE MA E' FUORI PORTA
TA. USCITA A OVEST."
2225 LET W=7: RETURN
2230 PRINT "SEI IN CIMA AD UNA S
CALA. UN CORRIDOIO VA A EST E
OVEST."
2235 LET E=69: LET W=67: LET D=118:
RETURN
2260 PRINT "SEI SU UNA SPORGENZA DEN
TRO UN BUCO. MOLTO PIU' IN ALTO
C'E UN ALTRO BUCO."
2264 IF R(2)=3 THEN PRINT "UNA FUNE
PENZOLA GIU'."
2265 RETURN
2310 RETURN
3000 PRINT "'LA PORTA E' APERTA.":
RETURN
3100 IF N=0 THEN GO TO 3130
3110 LET R(1)=N
3120 GO TO 30
3130 PRINT INK 2;"DIREZIONE NON CON
SENTITA"
3140 GO TO 90
3200 IF E=0 THEN GO TO 3130
3210 LET R(1)=E
3220 GO TO 30
3300 IF S=0 THEN GO TO 3130
3310 LET R(1)=S
3320 GO TO 30
3400 IF W=0 THEN GO TO 3130
3410 LET R(1)=W
3420 GO TO 30
3500 IF U=0 THEN GO TO 3130
3510 LET R(1)=U
3520 GO TO 30
3600 IF D=0 THEN GO TO 3130
3610 LET R(1)=D
3620 GO TO 30

```

```

3700 RESTORE 9040: FOR F=1 TO 30:
READ C$: IF LEN A$-LEN C$-1<1
THEN GO TO 3715
3710 IF C$=A$(LEN A$-LEN C$ TO LEN A$
-1) AND O(F)=R(1) THEN GO TO 37
20
3715 NEXT F: GO TO 3770
3720 FOR G=1 TO 5
3730 IF C(G)=0 THEN GO TO 3750
3740 NEXT G: PRINT INK 2;"PORTI TRO
PPI OGGETTI.": GO TO 90
3750 LET O(F)=0: LET C(G)=F
3760 PRINT " O.K.": GO TO 90
3770 PRINT INK 3;"NON POSSO VEDERE
!": GO TO 90
3810 FOR F=1 TO 5: RESTORE 9040:
FOR G=1 TO C(F): READ C$:
NEXT G
3820 IF LEN A$-LEN C$<1 THEN GO TO 3
840
3830 IF A$(LEN A$-LEN C$ TO LEN A$-1)
=C$ THEN GO TO 3850
3840 NEXT F: GO TO 3870
3850 LET C(F)=0: LET O(G-1)=R(1)
3860 GO TO 3760
3870 PRINT INK 3;"NON HAI L'OGGETTO
RICHIESTO!": GO TO 90
3900 IF B$="SCALETTA" THEN GO TO 395
0
3910 LET D$="STAMPO": GO SUB 7000:
IF H=6 THEN GO TO 5790
3920 LET D$="LIQUIDO VERDE": GO SUB 7
000: IF H=6 THEN GO TO 5790
3930 PRINT INK 1;"HAI CREATO UNA
STELLA DALLO STAMPO": LET D$="
STAMPO": GO SUB 7000: LET C(H)=3
: LET D$="LIQUIDO VERDE":
GO SUB 7000: LET C(H)=0:
GO TO 90
3950 LET D$="LEGNO": GO SUB 7000:
LET N=H: IF H=6 THEN GO TO 579
0
3960 LET D$="CHIODI": GO SUB 7000:
LET S=H: IF H=6 THEN GO TO 579
0
3970 LET D$="MARTELLO": GO SUB 7000:
LET W=H: IF H=6 THEN GO TO 579
0
3980 PRINT INK 1;"HAI UNA SCALETTA.
": LET C(N)=0: LET C(S)=0: LET C
(W)=0: LET C(W)=2: GO TO 90
4000 IF R(1)=11 THEN GO TO 4050
4010 IF R(1)=37 THEN LET D$="CHIAVE
BLU": GO SUB 7000
4020 IF H=6 THEN PRINT "'TI SERVE LA
CHIAVE GIUSTA.": GO TO 90
4030 IF R(1)=37 THEN PRINT "'PORTA
CHIUSA.": LET D(7,2)=0: LET D(7,
1)=0: GO TO 90
4040 IF R(1)<>37 THEN GO TO 5790
4050 LET D$="CHIAVE ROSSA":
GO SUB 7000: IF H=6 THEN
PRINT "'TI SERVE LA CHIAVE GIUS

```

```

TA.": GO TO 90
4060 PRINT "PORTA CHIUSA.": LET D(5,
2)=0: LET D(5,1)=0: GO TO 90
4100 IF R(1)<>24 THEN GO TO 5790:
IF B$<>"SAPONE" THEN GO TO 5
790
4110 LET D$="SAPONE": GO SUB 7000:
LET F=H: IF H=6 THEN GO TO 579
0
4120 GO SUB 7020: LET N=H: IF H=6
THEN GO TO 5790
4130 GO SUB 7020: IF H=6 THEN
GO TO 5790
4140 PRINT INK 1;"IL SAPONE SI SC
IOGLIE IN UN LIQUIDO VERDE.":
LET C(N)=0: LET C(F)=0: LET C(H
)=4
4150 GO TO 90
4400 FOR F=1 TO LEN A$
4405 IF A$(F)=" " THEN LET P=F+1:
IF A$(P TO P+3)="STEL" THEN
GO TO 4420
4410 NEXT F: GO TO 5790
4420 IF R(1)<>118 THEN GO TO 5790
4430 IF B$<>"BUCO" THEN GO TO 5790

4440 LET D$="STELLA": GO SUB 7000:
IF H=6 THEN GO TO 5790
4450 LET C(H)=0: PRINT "LA STELLA SI
ADATTA PRECISAMENTEAL BUCO.SUBI
TO LA SCALA SI MUOVEE APPARE UNA
STANZA. ENTRI.": LET R(1)=35:
GO TO 30
4500 IF B$="PORTA" THEN GO TO 4570
4510 IF B$="GUARDAROBA" THEN GO TO 4
550
4520 IF B$<>"SESAMO" THEN GO TO 57
20
4530 IF R(1)<>48 AND R(1)<>119
THEN GO TO 5720
4540 IF R(1)=119 THEN LET D(9,1)=1:
PRINT "IL MURO SI APRE COME FO
SSE UNA PORTA, RIVELANDO UN P
ASSAGGIO SEGRETO.": LET R(1)=56:
LET EN=48: GO TO 90
4545 IF R(1)=48 THEN LET D(9,1)=1:
PRINT "IL MURO SI APRE COME FO
SSE UNA PORTA, RIVELANDO UN C
ORRIDOIO.": LET R(1)=56: LET L=1
19: LET EN=119: GO TO 90
4550 IF R(1)<>73 THEN GO TO 5720
4560 PRINT "IL GUARDAROBA SI APRE.":
LET EN=75: LET D(8,1)=1:
GO TO 90
4570 IF R(1)<>110 AND R(1)<>2
AND R(1)<>4 AND R(1)<>7
AND R(1)<>11 AND R(1)<>21
AND R(1)<>37 THEN GO TO 5720

4579 IF R(1)=11 AND D(4,2)=1 THEN
GO TO 4584
4580 IF R(1)=2 THEN LET D(1,2)=1:
LET EN=119: GO SUB 3000:
GO TO 90

```

```

4581 IF R(1)=4 THEN LET D(2,2)=1:
LET EN=5: GO SUB 3000: GO TO 90

4582 IF R(1)=110 AND D(10,1)=1 THEN
LET D(10,2)=1: LET D=76: LET EN
=76: GO SUB 3000: GO TO 90
4583 IF R(1)=11 THEN LET D(4,2)=1:
LET EN=13: GO SUB 3000: GO TO 9
0
4584 IF R(1)=11 AND D(5,1)=1 THEN
LET D(5,2)=1: LET EN=14:
GO SUB 3000: GO TO 90
4586 IF R(1)=21 THEN LET D(6,2)=1:
LET EN=22: GO SUB 3000: GO TO 9
0
4587 IF R(1)=37 AND D(7,1)=1 THEN
LET D(7,2)=1: GO SUB 3000:
GO TO 90
4588 IF R(1)=37 AND D(7,1)=0 THEN
PRINT "LA PORTA E' CHIUSA.":
GO TO 90
4589 IF R(1)=11 AND D(5,1)=0 THEN
PRINT "LA PORTA E' CHIUSA.":
GO TO 90
4590 PRINT "NON PUOI APRIRE.":
GO TO 90
4600 IF R(1)=98 AND B$="MUSICA"
THEN GO TO 4650
4610 IF B$<>"MEDAGLIONE" THEN
GO TO 5790
4620 LET D$="MEDAGLIONE D'ORO":
GO SUB 7000: IF H=6 THEN
GO TO 5790
4630 PRINT "LO SCRITTO E' TROPPO PIC
COLO.": GO TO 90
4650 PRINT INK 3;"LE NOTE SONO: DO,
MI,FA,LA."
4660 GO TO 90
4700 LET D$="LENTE": IF B$="SCOPA"
THEN LET D$=B$
4705 GO SUB 7000: IF H=6 THEN
GO TO 5790
4710 IF B$="SCOPA" AND R(1)=122
THEN PRINT "TI ALLUNGI CON
LA SCOPA E MANOVRI PER PREME
RE IL BOTTONE. IL NASTRO TRASP
ORTATORE SI ARRESTA SUBITO.":
LET D(7,2)=1: LET E=8: GO TO 90

4720 LET D$="MEDAGLIONE D'ORO":
GO SUB 7000: IF H=6 THEN
GO TO 5790
4730 IF B$="LENTE" THEN PRINT "LA
SCRITTA DICE: ~ PER TROVAREUN C
ORRIDOIO NASCOSTO, CERCA UN POST
O VUOTO E STRANO, GRIDA APRI
TI SESAMO E LA TUA FORTUNA CAMB
IERA'. ~": GO TO 90
4790 GO TO 5790

4800 IF B$="FRUTTA" AND R(1)=43
THEN CLS : PRINT INK 4;"MANGI
IL CIBO E TI SENTI BENE. MA,D'
UN TRATTO, TI SENTI STORDITOE TI

```

```

GIRA LA TESTA. TI DIMENI E SALT
FINCHE' CROLLI A TERRA. HAI F
ALLITO LA PROVA.": GO TO 9960
4810 IF B$="FRUTTA" AND R(1)=44
THEN PRINT "'MANGI IL CIBO E T
I SENTI BENE. CAPISCI COME LA
VITA POTRA' ESSERE PIU' GRADE
VOLE D'ORA IN POI. HAI INFATTI
MANGIATO LA FRUTTA DELLA VITA
, E LA TUA VITASTESEA SARA' ALLU
NGATA A MENO CHE TU NON COM
METTA DEGLI IMPERDONABILI ERR
ORI.": LET LLIFE=0: GO TO 90
4820 PRINT INK 1"'MANGIARE ? NON SI
PUO'.": GO TO 90
4900 LET D$="FUNES": GO SUB 7000: IF H
=6 THEN GO TO 5790
4905 IF R(2)=0 THEN PRINT "'LANCI L
A FUNE MA NON AGGANCI NULLA.":
GO TO 6300
4910 LET D$="GANCIO": GO SUB 7000:
IF H=6 THEN RETURN
4920 IF R(2)=1 AND R(1)<>126 THEN
PRINT "'LANCI LA FUNE MA NON
AGGANCI NULLA.": GO TO 90
4930 IF R(2)=1 AND R(1)=126 THEN
PRINT "'LANCI LA FUNE E L'UNCI
NO SI AGGANCIA A QUALCOSA DI
SICURO.": LET D$="FUNES":
GO SUB 7000: LET R(2)=3: LET C(
H)=0: LET D$="GANCIO": GO SUB 70
00: LET C(H)=0: GO TO 90
5000 IF R(1)=11 THEN GO TO 5050
5010 IF R(1)=37 THEN LET D$="CHIAVE
BLU": GO SUB 7000
5020 IF H=6 THEN PRINT "'TI SERVE LA
CHIAVE GIUSTA.": GO TO 90
5030 IF R(1)=37 THEN PRINT "'PORTA
CHIUSA.": LET D(7,1)=0: GO TO 90
5040 IF R(1)<>37 THEN GO TO 5790
5050 LET D$="CHIAVE ROSSA":
GO SUB 7000: IF H=6 THEN
PRINT "'TI SERVE LA CHIAVE GIUS
TA.": GO TO 90
5060 PRINT "'PORTA CHIUSA.": LET D(5,
1)=1: GO TO 90
5100 IF B$="VINO" AND R(1)=49 THEN
PRINT "'IL VINO E' FRESCO E TI
RISTORA MENTRE SCIVOLA LEGGERO
NELLA TUAGOLA. TUTTO SEMBRA MERA
VIGLIOSO QUANDO, DI COLPO, SVIEN
I.": PAUSE 500: LET R(1)=20:
CLS : PRINT "'TI RISVEGLI SU UN
DIVANO.": GO TO 90
5110 PRINT "'NON PUOI BERE.": GO TO 9
0
5200 IF R(1)<>51 THEN GO TO 5790
5210 LET D$="SCALETTA": GO SUB 7000:
IF H=6 THEN GO TO 5790
5220 PRINT INK 1; "'TI ARRAMPICHI SIN
O AL MARGINE DI UNA BOTOLA. LA SC
ALETTA CASCA LONTANO. IN ALTO
VEDI SOLO UN BUCO.": LET R(1)=
126: LET C(H)=0: GO TO 90
5300 GO TO 5400
5399 STOP
5400 IF EN=0 THEN GO TO 5420
5410 LET R(1)=EN: GO TO 30
5420 PRINT INK 2; "'NON PUOI USARE EN
TRO O DENTRO.": GO TO 90
5500 GO TO 5600
5600 IF L=0 THEN GO TO 5620
5610 LET R(1)=L: GO TO 30
5620 PRINT INK 2; "'NON PUOI USARE ES
CO O FUORI.": GO TO 90
5700 IF B$="TAPPETO" THEN GO TO 5750
5710 IF B$="SCAFFALE" THEN GO TO 573
0
5715 IF R(1)=10 AND B$="QUADRI"
THEN GO TO 5791
5720 PRINT INK 2; "'NON PUOI FARE CIO
'CHE HAI DETTO.": GO TO 90
5730 IF R(1)<>55 THEN GO TO 5790
5740 LET EN=58: LET R(1)=57: PRINT "'
LO SCAFFALE SI MUOVE RIVELANDO
UNA STANZA SEGRETA.": GO TO 90
5750 IF R(1)<>110 THEN GO TO 5790
5760 PRINT "'IL TAPPETO SI SPOSTA RIV
ELANDO UNA BOTOLA NEL PAVIMENTO
."
5770 LET D(10,1)=1: GO TO 90
5790 PRINT INK 2; "'NON SI PUO'.":
GO TO 90
5791 PRINT "'DIETRO UN QUADRO SI TROV
A UNA CASSAFORTE MA E' CHIUSA.
PROVA AD APRIRLA SE CREDI DI C
ONOSCKERELA COMBINAZIONE.":
GO TO 90
5799 STOP
5800 CLS : PRINT "'RICORDA CHE CI SO
NO 4 PARTI DA SALVARE."
5810 SAVE "ELENA1" DATA O()
5820 SAVE "ELENA2" DATA R()
5830 SAVE "ELENA3" DATA C()
5840 SAVE "ELENA4" DATA D()
5850 PRINT "' O.K."
5860 GO TO 90
5900 CLS : PRINT "CARICAMENTO: VIA CO
N IL NASTRO"
5910 LOAD "ELENA1" DATA O(): LOAD "EL
ENA2" DATA R(): LOAD "ELENA3"
DATA C(): LOAD "ELENA4" DATA D(
)
5920 GO TO 5850
6000 GO TO 30
6100 PRINT INK 1; "'STAI PORTANDO:"
6110 LET H=1: FOR F=1 TO 5: IF C(F)=0
THEN GO TO 6140
6120 LET H=0
6130 RESTORE 9040: FOR G=1 TO C(F):
READ C$: NEXT G: PRINT INK 3;C
$
6140 NEXT F: IF H=1 THEN PRINT
INK 2; "NIENTE."

```

```

6150 GO TO 90
6200 IF R(1)=24 THEN PRINT "'PERCHE'
NON PROVI A SCALDARE QUALCOS
A ?"
6210 IF R(1)=118 THEN PRINT "'METTI
UNA STELLA NEL BUCO."
6220 IF R(1)=51 THEN PRINT "'PROVA A
D ARRAMPICARTI ATTRAVERSOIL BUCO
AIUTANDOTI CON QUALCOSA."
6230 IF R(1)=34 THEN PRINT "'E SE PR
OVASSI AD ALLACCIARE FRA LORO UN
GRAN NUMERO DI LENZUOLI?"
6250 PRINT "'HAI PROVATO A FARE UNO S
TAMPO ?"
6280 IF R(1)<>51 AND R(1)<>118
AND R(1)<>24 THEN PRINT "'NO
N HO NESSUN'ALTRA IDEA."
6290 GO TO 90
6300 LET D$="FUNES": GO SUB 7000
6310 LET D$="GANCIO": GO SUB 7000
6320 IF H=6 THEN GO TO 5790
6330 LET R(2)=1
6340 PRINT "'PERO'....RIPROVA!":
GO TO 90
6400 IF R(1)<>126 OR R(2)<>3
THEN GO TO 5790
6410 CLS : PRINT "'TI ARRAMPICHI SULL
A FUNE SINO IN CIMA.": LET R(1
)=113: GO SUB 2130: GO TO 90
6500 GO TO 4400
6600 IF R(1)<>34 THEN GO TO 5790
6605 FOR F=1 TO 5: LET D$="LENZUOLO":
GO SUB 7000: IF H=6 THEN
GO TO 5790
6610 NEXT F
6620 PRINT "'HAI FATTO UNA FUNE ANCHE
SE PER ORA NON SAI COME USARLA.
": LET C(1)=1: FOR F=2 TO 5:
LET C(F)=0: NEXT F: GO TO 90
6700 IF R(1)=10 THEN PRINT INK 1"'L
A CASSAFORTE SI APRE E NE ESCE U
N LINGOTTO D'ORO. LA CASSAFORTES
VANISCE NEL NULLA.": LET O(14)=1
0: GO TO 90
6710 PRINT "'COSA INTENDI ?": GO TO 9
0
7000 FOR H=1 TO 5
7010 RESTORE 9040: FOR G=1 TO C(H):
READ C$: NEXT G: IF C$=D$
THEN RETURN
7020 NEXT H: RETURN
8999 STOP
9000 DATA "N","E","S","O","NORD","EST
","SUD","OVEST","ALTO","BASSO","
A","B"
9002 DATA "PRENDO","LASCIO","FACCIO",
"BLOCCO","SCALDO","ALLACCIO","LE
GO","METTO","APRO","LEGGO","USO"
9004 DATA "MANGIO","GETTO","SBLOCCO",
"BEVO","SALGO","DENTRO","ENTRO",
"FUORI","ESCO","MUOVO"
9006 DATA "SALVA","CARICA","GUARDO","
LISTA","INVENTARIO"

```

```

9008 DATA "AIUTO","2800"
9010 DATA 1,2,3,4,1,2,3,4,5,6,5,6,7,8
,9,10,11,913,13,14,15,16,17,18,1
9,20,21,22,23,24,25,26,27,28,29,
30,31,31,32,33
9020 DATA "1","2","3","4","5","6","7"
,"8","9","10","11","12","13","14
","15","16","17","18","19","20",
"21","22","23","24","25","26","2
7","28","29","30","31","32","121
3","2213","1423","913","33"
9040 DATA "FUNES","SCALETTA","STELLA",
"LIQUIDO VERDE","LIBRO","SCATOLA
DI SIGARI","CHIODI","MARTELLINO",
"LEGNO","MEDAGLIONE D'ORO","LENZ
UOLO","LENZUOLO","LENZUOLO"
9042 DATA "LINGOTTO D'ORO","LENZUOLO"
,"LENZUOLO","LENZUOLO","LENZUOLO
","STAMPO","LENZUOLO","SCOPA","R
UBINO","GANCIO","CHIAVE BLU","CH
IAVE ROSSA"
9045 DATA "LENTE","LEGNO","SAPONE","S
APONE","SAPONE"
9050 DATA 0,0,0,0,14,13,38,38,46,59,6
1,64,70,65,78,81,82,83,111,111,3
5,76,75,14,58,18,38,5,30,106
9930 DEF FN A()=((65536*PEEK 23674+25
6*PEEK 23673+PEEK 23672)/50)
9940 POKE 23674,0: POKE 23673,0:
POKE 23672,0: RETURN
9950 IF R(1)=54 THEN CLS : PRINT "'G
LI OCCHI MALVAGI APPARTENGONO A
D UNA ENORME VEDOVA NERA. ESSI S
I MUOVONO VERSO DI TE COME NON F
OSSERO LEGATI AL CORPO. LE SUE M
ANDIBOLE SI APRONO SU DI TE E S
TRINGONO MORTALMENTE LE TUE B
RACCIA. IL VELENO AGISCE R
APIDAMENTE E MUORI."
9960 IF R(1)=17 THEN CLS : PRINT
INK 4;"SEI STATO TROPPO LENTO
! LO STRANO ODORE ERA UN GAS
LETALE E SEI MORTO PER QUESTO.
"
9970 IF FN A(>2000 THEN CLS :
PRINT INK 4"'NON HAI MANGIATO
LA FRUTTA DELLAVITA E L'AMBIENTE
OSTILE CHE TI HA CIRCONDATO PER
TUTTO QUESTO TEMPO TI HA SO
TTRATTO TUTTA L'ENERGIA: SEI MO
RTO."
9980 PRINT INK 1"' SEI STATO S
FORTUNATO!"
9990 FOR F=1 TO 30: BEEP .01,F:
NEXT F: FOR F=30 TO 1 STEP -1:
BEEP .01,F: NEXT F: FOR F=1
TO 30 STEP 2: BEEP .01,F:
NEXT F: FOR F=30 TO 1 STEP -2:
BEEP .01,F: NEXT F: GO TO 9990

```

Ridefinizione dei tasti

di Paolo Maffei

Questo programma, scritto completamente in linguaggio macchina, vi permette di ridefinire la posizione di tutti i tasti del vostro Spectrum, rendendo quindi sempre più semplice e veloce la stesura dei vostri programmi.

Potrete ridefinire la posizione sulla tastiera delle semplici lettere, della punteggiatura, dei caratteri grafici oppure dei comandi (PRINT, BEEP, READ, ecc.).

La ridefinizione dei tasti va inserita in una linea REM all'inizio del programma su cui state lavorando, utilizzando la seguente sintassi: numero di linea (è importante che questo sia il più basso possibile), il comando REM, il carattere che deve essere sostituito, il simbolo "." ed infine il carattere che dovrà prendere posto sulla tastiera nella nuova posizione.

Un esempio servirà a chiarire il semplice meccanismo:

1 REM p;

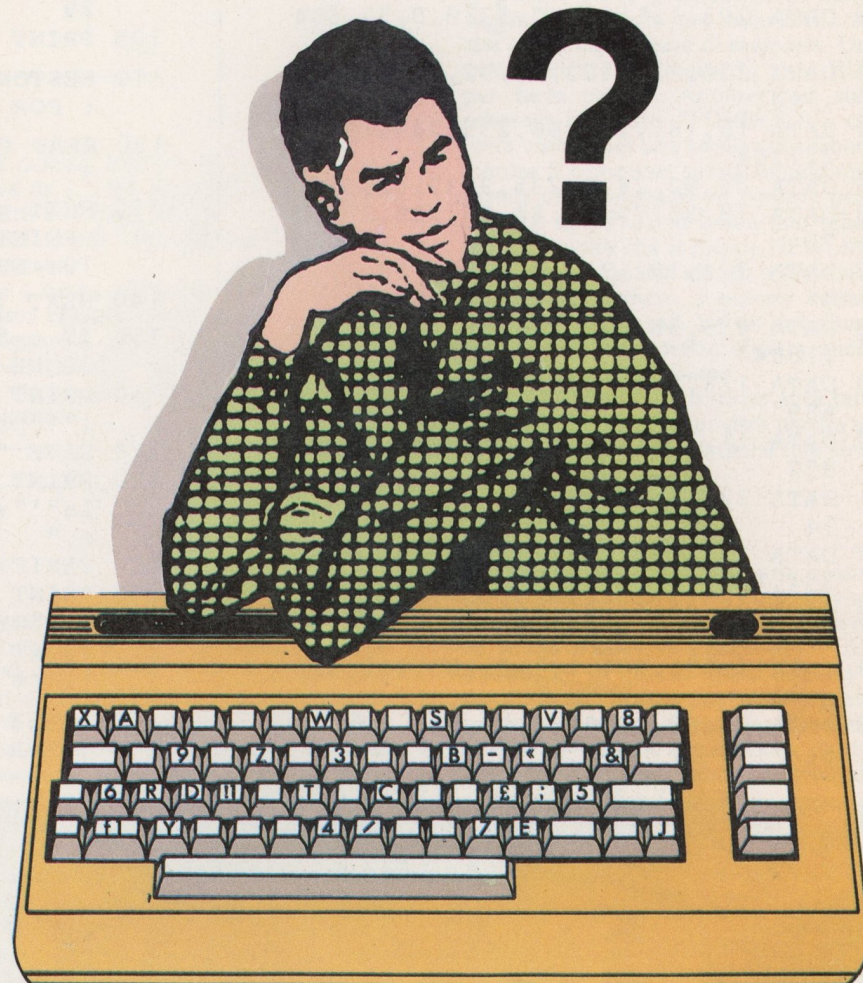
Questa linea permette di ottenere il simbolo "." alla pressione del solo tasto p (se al posto della p aveste lasciato uno spazio, avreste ottenuto il simbolo " ", alla pressione del tasto SPACE, rendendo molto più comoda la stesura, per esempio, di lunghe istruzioni DATA). Il simbolo "#" posto alla fine della linea permetterà l'esecuzione automatica del comando.

Esempio:

1 REM ! : PAPER 7: INK 0: CLS: BORDER 7 #

Questo comando resetta tutti i colori ai valori originali (carta bianca, cornice bianca, inchiostro nero), premendo solamente i tasti SYMBOL SHIFT e 1 (ciò che avrebbe dovuto corrispondere al simbolo !); questo è molto comodo quando si è alle prese con programmi con tali variazioni di colori da rendere spesso illeggibile il listato.

Ancora un esempio:



1 REM NEW : INPUT A\$: IF A\$ = "SI" THEN NEW

Questa linea salva il programma su cui state lavorando da una accidentale e indesiderata pressione del comando NEW. Se ciò dovesse accadere, vi verrebbe richiesta una conferma prima di cancellare ogni cosa.

Caricamento del programma

Il linguaggio macchina è contenuto in linee DATA nel programma BASIC. Notate che la nona cifra di ogni linea DATA è la somma delle 8 precedenti: questo artificio viene utilizzato per correggere velocemente eventuali errori di

copiatura.

Copiate il listato e, per sicurezza, salvatelo su un nastro; dopo questa operazione potete dare il RUN, attendete qualche secondo e, se avete fatto qualche errore di copiatura, questo vi verrà comunicato con un messaggio; dovrete quindi apportare la necessaria modifica e, ancora una volta, date il RUN: se non ci sono altri errori, dovrete, seguendo sempre le istruzioni che compaiono di volta in volta sul video, registrare sul nastro in linguaggio macchina, verificarlo ed infine cancellare definitivamente il programma BASIC.

Per utilizzare in seguito questa routine dovrete battere:

CLEAR 32347 (16K)
CLEAR 65128 (48K)

seguito da

LOAD "" CODE

Per attivare il programma dovrete utilizzare il comando

RANDOMIZE USR 32479 (16K)
RANDOMIZE USR 65260 (48K)

Per disattivarlo

RANDOMIZE USR 32486 (16K)
RANDOMIZE USR 65267 (48K)

Per disattivare momentaneamente una istruzione di ridefinizione sarà comunque sufficiente spostare la linea REM dall'inizio del programma alla sua fine.

Ridefinizione dei tasti versione per ZX Spectrum 16K/48K

```
10 DATA 255,243,229,213,197,245,205,107,1694
11 DATA 126,241,193,209,225,251,201,253,1699
12 DATA 203,1,110,200,33,0,0,57,604
13 DATA 235,237,123,61,92,225,1,127,1101
14 DATA 16,167,237,66,235,249,192,42,1204
15 DATA 83,92,24,2,235,9,35,35,515
16 DATA 78,35,70,35,84,93,126,254,775
17 DATA 234,192,35,58,8,92,190,32,841
18 DATA 235,35,126,254,58,32,229,35,1004
19 DATA 126,254,13,40,223,11,11,11,689
20 DATA 11,197,229,42,91,92,205,85,952
21 DATA 22,19,237,83,91,92,35,235,814
22 DATA 225,193,237,176,235,43,126,254,1489
23 DATA 35,40,8,205,29,17,253,203,790
24 DATA 1,174,201,1,1,0,205,232,815
25 DATA 25,62,13,50,8,92,253,203,706
```

```
26 DATA 1,238,201,62,40,237,71,237,1087
27 DATA 94,201,62,62,237,71,237,86,1050
28 DATA 201,0,0,0,0,0,0,0,201
100 LET a=32348: IF PEEK 23732+256*PEEK 23733>32767 THEN LET a=65129
105 PRINT "ATTENDERE,PREGO."
110 RESTORE : FOR f=0 TO 18: LET t=0 : FOR g=0 TO 7
120 READ d: POKE a+f*8+g,d: LET t=t+d
130 NEXT g: READ d: IF d<>t THEN PRINT "ERRORE nella linea ";f+10: STOP
140 NEXT f
150 IF a=65129 THEN POKE 65261,9: POKE 65136,120: POKE 65137,254
160 PRINT "Preparati a registrare il programma ~RID.TASTI~"
170 SAVE "RID.TASTI"CODE a,150
180 PRINT "Riavvolgi il nastro per la verifica della registrazione."
190 VERIFY "RID.TASTI"CODE a,150
200 PRINT "ON : RANDOMIZE USR ";a+131"OFF: RANDOMIZE USR ";a+138
210 PRINT "Premi un tasto per cancellare il programma BASIC."
220 PAUSE 0
230 CLEAR a-1: NEW
```

Avete mai "pokato" nel byte 23624? No? Beh, perché non provate? Ricordatevi solo che, in caso di difficoltà, basterà digitare (magari alla cieca!) POKE 23624,56. Se siete privi di fantasia, potete copiare questo semplice programma:

```
10 FOR i=0 TO 255
20 POKE 23524,i
30PRINT # 0; AT0,0; "Ma guarda che roba... ma si può? È proprio qualcosa di buffo!!!"
40 NEXT i
50 POKE 23624,56
```


Tombola 1k

di **Guido Bertolotti**

Descrizioni delle variabili

A\$	è il nostro sacchetto di numeri. Prima viene riempito con i caratteri il cui codice va da 1 a 90, poi i caratteri vengono mescolati e infine estratti uno dopo l'altro
B	variabile di loop. È riciclata più volte per non riempire la memoria con troppe variabili
D	anche questa variabile è usata più volte: alla linea 90 le viene assegnato un valore casuale compreso fra 1 e B; alla linea 160 le viene assegnato il valore del numero estratto
D\$	è usata in un'istruzione di INPUT alla linea 140: se non viene inserita una "T", viene estratto un numero; altrimenti comincia una nuova partita

Descrizione del programma

Linee	Commento
10-20 30-110	presentazione inizializzazione. È composta da due parti: la prima (linee 40-70) genera una stringa (A\$) i cui caratteri hanno, in ordine, codici da 1 a 90. La seconda parte (linee 80-110) provvede a mescolare casualmente i caratteri di questa stringa. Il mescolamento utilizza le funzioni di slicing ("affettamento") delle stringhe, e funziona in pratica così: al primo passaggio di un ciclo viene scelto a caso uno dei 90 caratteri di A\$, estratto dalla posizione che occupa e trasferito in fondo alla stringa. Al secondo passaggio viene scelto a caso uno dei primi 89 caratteri (il 90°, cioè quello appena trasferito, non viene più toccato) e trasferito in fondo alla stringa. Al terzo passaggio il cursore in trasferimento sarà uno dei primi 88, e così via, per un totale di 90 passaggi. Questo sistema di utilizzare una stringa al posto di un vettore per memorizzare dei numeri presenta due grossi vantaggi: 1) si risparmia memoria, perché ogni elemento della stringa (carattere) occupa un byte, mentre ogni elemento di un vettore (numero) occupa cinque byte; 2) il mescolamento dei caratteri della stringa, grazie alle funzioni di slicing, risulta più agevole e veloce del mescolamento dei numeri di un vettore. Il limite è, naturalmente, che in una stringa si possono memorizzare, sotto forma di codici di caratteri, solo numeri interi compresi fra 0 e 255
130-180	estrazione dei numeri. Questo loop legge, uno dopo l'altro, i codici dei caratteri della stringa A\$, che sono già mescolati casualmente: i numeri estratti sono pertanto in ordine casuale. Non esiste il pericolo che un numero venga estratto più di una volta, perché il carattere corrispondente compare in A.dol\$ una volta sola

Questo breve programma (si carica in 15 secondi) gira sullo ZX81 anche senza espansione di memoria. Una volta caricato è necessario dare RUN per farlo partire. Appare così sullo schermo la scritta "TOMBOLA", che resta visibile per una ventina di secondi, mentre il computer svolge alcune operazioni di inizializzazione. Appare poi il cursore di INPUT in fondo allo schermo: ogni volta che si preme NEWLINE (si inserisce cioè una stringa nulla) viene estratto un numero. I numeri estratti vengono visualizzati in un tabellone, mentre l'ultimo estratto appare anche qualche riga più in basso.

Quando il fortunato di turno avrà fatto tombola, per giocare un'altra partita sarà sufficiente inserire, anziché una stringa nulla, la lettera "T".
Buon divertimento.

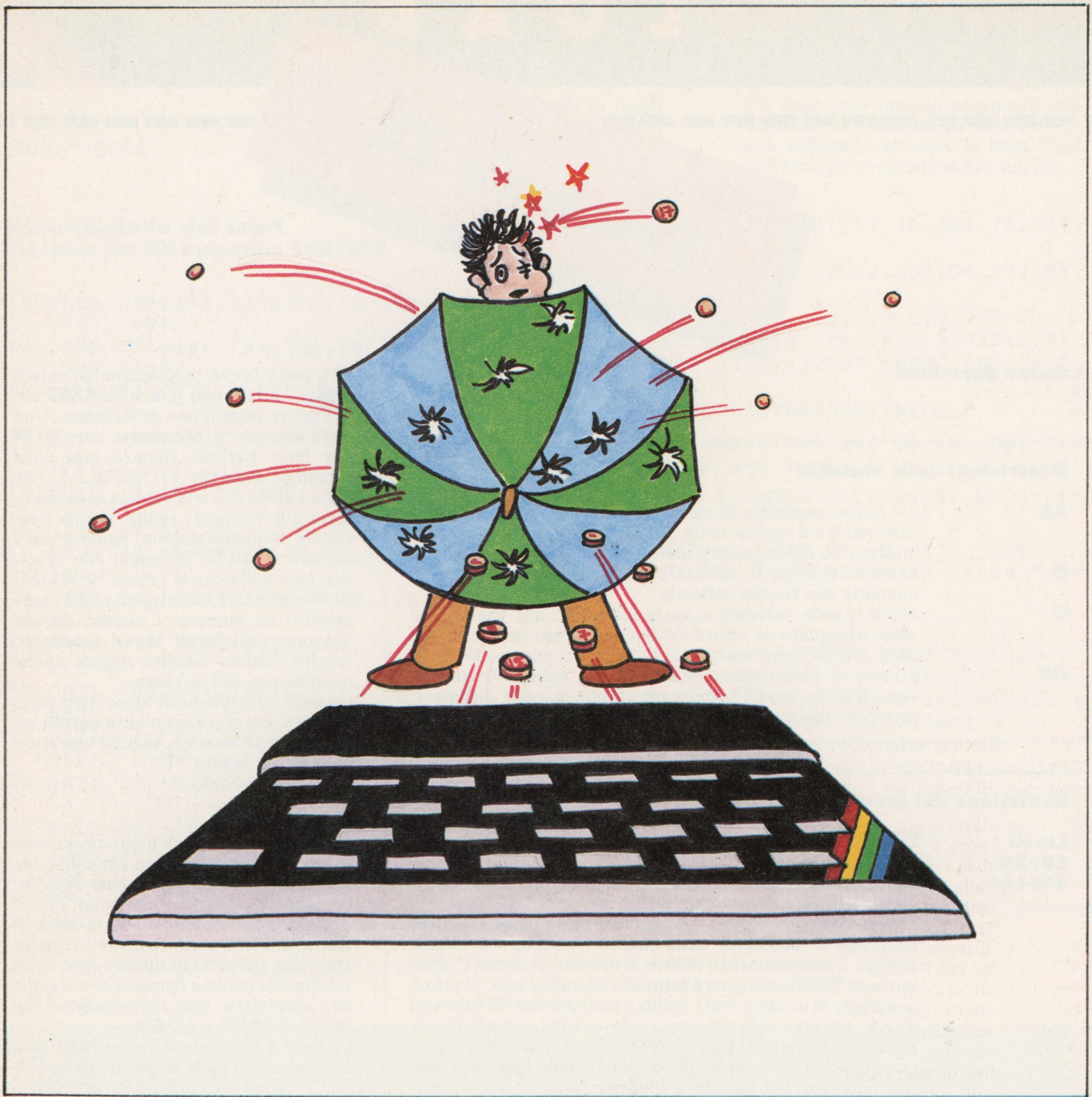
Nota bene

Chi ha l'espansione di memoria e vuole mantenere il programma caricabile anche su ZX81 senza espansione deve dare, prima di salvarlo, il comando: POKE 16389,68. In questo modo la zona di memoria contenente i caratteri visualizzati sullo schermo (il display file) viene ridotta alle minime dimensioni e si ottiene, oltretutto, una abbreviazione dei tempi di SAVE e LOAD.

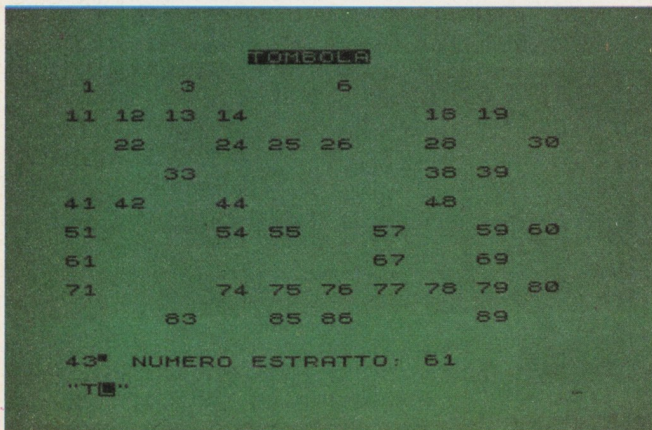
La linea 170 contiene un carattere grafico racchiuso fra le virgolette di un CODE " ". Esso è CHR\$ 10 (si ottiene con il cursore in modo GRAPHICS e SHIFT S): fate attenzione a non confonderlo con un CHR\$ 3 (GRAPHICS e SHIFT 7).

Ricopiate il programma esattamente com'è, *specialmente la linea 170*. Se esso si blocca con un errore 4 (fine memoria), significa che nel ricopiarlo vi siete lasciati sfuggire qualche carattere di troppo. Se non riuscite a trovarlo ed eliminarlo, cancellate eventualmente la linea 20.

I vari NOT PI, SGN PI, INT PI, VAL "90", CODE "+", che compaiono nel listato, non sono altro che modi complicati per dire, rispettivamente, 0, 1, 3, 90, 21. Sono utilizzati al posto dei numeri scritti esplicitamente, perché consentono di risparmiare memoria e permettono a "Tombola 1K" di girare, appunto, anche con un solo Kilobyte di RAM (avanza solo 1 byte!).



Tombola 1K
 versione per ZX81 1K



```

10 CLS
20 PRINT TAB PI*PI;" TOMBOLA"
30 RAND
40 LET A$=""
50 FOR B=SGN PI TO VAL "90"
60 LET A$=A$+CHR$ B
70 NEXT B
80 FOR B=VAL "90" TO SGN PI ST
EP -SGN PI
90 LET D=INT (RAND*B+SGN PI)
100 LET A$(D TO )=A$(D+SGN PI T
0 )+A$(D)
110 NEXT B
130 FOR B=SGN PI TO VAL "90"
140 INPUT D$
150 IF D$="T" THEN RUN
160 LET D=CODE A$(B)
170 PRINT AT CODE "+",NOT PI;B;
" NUMERO ESTRATTO: ";D;" ";AT U
AL "INT (D/10.1)*2+2", (D-INT (D/
10.1)*CODE "°")*INT PI+(PI*PI>D)
-INT PI;D
180 NEXT B
  
```

Spectrum renumber

di **Marco Broglia**
e **Davide Conti**

Alcune routine della ROM dello Spectrum utilizzate in "Spectrum renumber"

Indir dec	Indir hex	Nome routine ROM	Descrizione
16	00 10	PRINT A CHARACTER	Il registro A contiene il codice del carattere che deve essere stampato.
3082	0C 0A	MESSAGE PRINTING	Questa subroutine è usata per stampare messaggi e tokenizzati. Il registro A contiene il "numero d'ingresso" del messaggio o tokenizzato in una tabella. La coppia di registri DE contiene l'indirizzo di base della tabella.
3435	0D 6B	CLS COMMAND	Cancella l'intero display.
4764	12 9C	MAIN EXECUTION	Il ciclo principale si estende dalla locazione 12A2 hex (4770 dec) alla locazione 15AE hex (5550 dec) e controlla l'"editing mode", l'esecuzione di comandi diretti e la produzione di riporti.
5633	16 01	CHAN-OPEN	La subroutine è chiamata con il registro A contenente un numero di flusso valido. In base al flusso di dati selezionato verrà aperto un particolare canale.
5717	16 55	MAKE-ROOM	Questa è una subroutine molto importante. È chiamata in molte occasioni per aprire un'area. In tutti i casi la coppia di registri HL punta alla locazione successiva al punto dove lo spazio è richiesto e la coppia di registri BC contiene la lunghezza dello spazio voluto.
6326	18 B6	NUMBER	Se il registro A contiene il codice 0E hex (14 dec) la coppia di registri HL conterrà l'indirizzo del byte successivo al formato floating-point.
6510	19 6E	LINE-ADDR	Per un dato numero di linea, questa subroutine ritorna con l'indirizzo di partenza della stessa linea o della prima linea successiva nella coppia di registri HL, e l'inizio della linea precedente nella coppia di registri DE. Se il numero di linea viene trovato il flag Z sarà attivato.

Renumber Assembler

Come già annunciato pubblichiamo il listato Assembler del programma "Spectrum renumber" - di Marco Broglia e Davide Conti - il cui listato BASIC è stato pubblicato su SUPERSINC numero 3, datato marzo 1985.

Ci scusiamo con gli autori e con i lettori del ritardo intercorso tra la pubblicazione delle due parti dell'articolo, dovuto a problemi tecnici. Il disassemblato, caratterizzato dall'output abbastanza atipico, è stato ottenuto con il programma monitor/assembler della Picturesque.

Nota: 1 Ciclo di clock = 0.00000025 secondi = 250 nanosecondi. Quanto sono mostrati due possibili tempi di esecuzione (cioè 7/12), ciò indica che il numero dei cicli di clock dipende dai flag delle condizioni. Il primo tempo vale per "condizione non verificata", mentre il secondo vale per "condizione verificata".

6536	19 88	FIND EACH STATEMENT	Questa subroutine può essere usata per cercare il "D"esimo statement in una linea BASIC, ritornando con la coppia di registri HL contenente l'indirizzo della locazione precedente all'inizio dello statement e il flag Z attivato.
6584	19 B8	NEXT-ONE	Questa subroutine può essere usata per cercare la linea successiva nell'area programma o la variabile successiva nell'area variabili.
6632	19 E8	RECLAIMING	La coppia di registri HL punta alla prima locazione da cancellare e la coppia di registri BC contiene il numero dei byte che devono essere cancellati.
6683	1A 1B	REPORT AND LINE NUMBER PRINTING	La coppia di registri BC contiene il numero da stampare. Qualsiasi valore oltre 27 0Fhex (9999 dec) non sarà tuttavia stampato correttamente.
12457	30 A9	HL=HLxDE	La subroutine è chiamata per compiere la moltiplicazione a 16 bits qualsiasi overflow riscontrato provocherà un ritorno alla subroutine.

Tabella delle istruzioni Assembly utilizzate in "Spectrum renumber"

Num	Istruzioni Assembly	Cicli di clock	Byte	Frequenza d'uso	Totale byte
01	ADD A, data	7	2	3	6
02	ADD HL, rp	11	1	13	13
03	AND reg	4	1	7	7
04	CALL label	17	3	40	120
05	CP data	7	2	20	40
06	CP (HL)	7	1	2	2
07	CPIR	20/16	2	1	2
08	CP reg	4	1	4	4
09	DEC reg	4	1	1	1
10	DEC rp	6	1	6	6
11	DEFB data	0	1	14	14
12	DEFM string	0	var	10	64
13	DEFS n	0	n	6	10
14	DI	4	1	2	2
15	DJNZ disp	8/13	2	1	2
16	EI	4	1	3	3
17	END	0	0	1	0
18	EQU data	0	0	4	0
19	EX DE, HL	4	1	11	11
20	EX (SP), HL	19	1	1	1
21	INC reg	4	1	4	4
22	INC rp	6	1	38	38
23	JP cond, label	10	3	12	36
24	JP (IX)	8	2	1	2
25	JP label	10	3	11	33
26	JR C, disp	7/12	2	5	10
27	JR disp	12	2	10	20
28	JR NC, disp	7/12	2	4	8
29	JR NZ, disp	7/12	2	12	24
30	JR Z, disp	7/12	2	12	24
31	LD A, (addr)	13	3	2	6
32	LD (addr), A	13	3	4	12
33	LD (addr), HL	16	3	5	15
34	LD (addr), rp	20	4	1	4
35	LD A, (DE)	7	1	6	6
36	LD (DE), A	7	1	4	4
37	LD dst, src	4	1	22	22
38	LD HL, (addr)	16	3	13	39
39	LD (HL), reg	7	1	8	8

Proposte per "Spectrum renumber" versione avanzata

1) permette l'uso del break in fase di immissione parametri.

2) rimuovere dal listato eventuali codici di attributi, gli spazi di testa e quelli di coda associati ai salti.

3) segnare i salti calcolati prima di rinumerare.

4) nella versione attuale, prima di richiedere l'immissione di ciascuno dei 4 parametri (From, To, Start, Step), lo schermo viene preventivamente cancellato; sicché sono necessarie 4 istruzioni di CLS.

Si propone di inserire una sola istruzione di CLS in corrispondenza della routine "INPUT".

5) l'attuale versione contiene un bug, che, sebbene riguarda situazioni alquanto rare, può portare a delle rinumerazioni indensiderate.

Se la destinazione del salto è espressa mediante un numero con più di 4 cifre, non sempre verrà dato il messaggio "Impossibile Renumbering", ma anzi il salto verrà rinumerato con un valore non precisato.

Ciò è dovuto ad una imprecisione presente nella parte diagnostica del programma, la quale si limita a verificare che la destinazione venga indicata esclusivamente da cifre ASCII, ma senza effettuare controlli sul loro numero.

40	LDIR		20/16	2	1	2
41	LD	IX, data	14	4	4	16
42	LD	reg, data	7	2	24	48
43	LD	reg, (HL)	7	1	28	28
44	LD	rp, (addr)	20	4	8	32
45	LD	rp, data	10	3	26	78
46	NEG		8	2	1	2
47	ORG	data	0	0	1	0
48	POP	rp	10	1	22	22
49	PUSH	IX	15	2	1	2
50	PUSH	rp	11	1	11	11
51	RES	b, (HL)	15	2	2	4
52	RET		10	1	3	3
53	RET	cond	5/11	1	1	1
54	RST	n	11	1	7	7
55	SBC	HL, rp	15	2	8	16
56	SUB	data	7	2	4	8
57	SUB	reg	4	1	1	1
58	XOR	reg	4	1	6	6

483 910

Spectrum Renumber disassemblato

Indirizzo
decimale

Indirizzo
esadecimale

64458	FB	CA				
23635	5C	53				
23627	5C	4B				
23560	5C	08				
23611	5C	3B				
64458	FB	CA	AF			
64459	FB	CB	CD	01	16	
64462	FB	CE	CD	6B	0D	
64465	FB	D1	11	10	FF	
64468	FB	D4	AF			
64469	FB	D5	CD	0E	FD	
64472	FB	D8	20	05		
64474	FB	DA	2A	53	5C	
64477	FB	DD	18	06		
64479	FB	DF	CD	6E	19	
64482	FB	E2	C2	B1	FC	
64485	FB	E5	22	00	FF	
64488	FB	E8	CD	6B	0D	
64491	FB	EB	11	10	FF	
64494	FB	EE	3E	01		
64496	FB	F0	CD	0E	FD	
64499	FB	F3	28	0E		
64501	FB	F5	CD	6E	19	
64504	FB	F8	C2	B1	FC	
64507	FB	FB	CD	B8	19	
64510	FB	FE	1A			
64511	FB	FF	FE	40		
64513	FC	01	38	0C		
64515	FC	03	21	10	27	
64518	FC	06	E5			
64519	FC	07	2A	4B	5C	
64522	FC	0A	22	02	FF	
64525	FC	0D	18	17		
64527	FC	0F	ED	53	02	FF
64531	FC	13	1A			
64532	FC	14	67			
64533	FC	15	13			
64534	FC	16	1A			
64535	FC	17	6F			
64536	FC	18	1B			
64537	FC	19	E5			
64538	FC	1A	2A	00	FF	
64541	FC	1D	A7			
64542	FC	1E	ED	52		

PROG
VARS
LASTK
FLAGS
MAIN

PRFRM

NTDEF

LDFRM
PRTO

EXIST

LIMIT

LTDO

ORG
EQU
EQU
EQU
EQU
XOR
CALL
CALL
LD
XOR
CALL
JR
LD
JR
CALL
JP
LD
CALL
LD
LD
CALL
JR
CALL
JP
CALL
LD
CP
JR
LD
PUSH
LD
LD
JR
LD
LD
LD
INC
LD
LD
DEC
PUSH
LD
AND
SBC

OFBCAH
5C53H
5C4BH
5C08H
5C3BH
A
1601H
0D6BH
DE, MSSG
A
INPUT
NZ, NTDEF
HL, (PROG)
LDFRM
196EH
NZ, PRMER
(DEP1), HL
OD6BH
DE, MSSG
A, 1
INPUT
Z, LIMIT
196EH
NZ, PRMER
19B8H
A, (DE)
64
C, LTDO
HL, 10000
HL
HL, (VARS)
(DEP2), HL
CNTLN
(DEP2), DE
A, (DE)
H, A
DE
A, (DE)
L, A
DE
HL
HL, (DEP1)
A
HL, DE

64544	FC 20	38	04				JR	C, CNTLN
64546	FC 22	E1					POP	HL
64547	FC 23	C3	B1	FC			JP	PRMER
64550	FC 26	01	FF	FF		CNTLN	LD	BC, -1
64553	FC 29	ED	5B	00	FF		LD	DE, (DEP1)
64557	FC 2D	1A					LD	A, (DE)
64558	FC 2E	FE	40				CP	64
64560	FC 30	30	11				JR	NC, EMPTY
64562	FC 32	C5				NXTON	PUSH	BC
64563	FC 33	2A	02	FF			LD	HL, (DEP2)
64566	FC 36	A7					AND	A
64567	FC 37	ED	52				SBC	HL, DE
64569	FC 39	28	0A				JR	Z, PRSTR
64571	FC 3B	EB					EX	DE, HL
64572	FC 3C	CD	B8	19			CALL	19B8H
64575	FC 3F	C1					POP	BC
64576	FC 40	03					INC	BC
64577	FC 41	18	EF				JR	NXTON
64579	FC 43	03				EMPTY	INC	BC
64580	FC 44	C5					PUSH	BC
64581	FC 45	CD	6B	0D		PRSTR	CALL	0D6BH
64584	FC 48	11	10	FF			LD	DE, MSSG
64587	FC 4B	3E	02				LD	A, 2
64589	FC 4D	CD	0E	FD			CALL	INPUT
64592	FC 50	20	03				JR	NZ, LDSTR
64594	FC 52	21	0A	00			LD	HL, 10
64597	FC 55	22	04	FF		LDSTR	LD	(DEP3A), HL
64600	FC 58	ED	5B	00	FF		LD	DE, (DEP1)
64604	FC 5C	2A	53	5C			LD	HL, (PROG)
64607	FC 5F	A7					AND	A
64608	FC 60	ED	52				SBC	HL, DE
64610	FC 62	28	1A				JR	Z, PRSTP
64612	FC 64	EB					EX	DE, HL
64613	FC 65	7E					LD	A, (HL)
64614	FC 66	23					INC	HL
64615	FC 67	6E					LD	L, (HL)
64616	FC 68	67					LD	H, A
64617	FC 69	CD	6E	19			CALL	196EH
64620	FC 6C	EB					EX	DE, HL
64621	FC 6D	7E					LD	A, (HL)
64622	FC 6E	23					INC	HL
64623	FC 6F	6E					LD	L, (HL)
64624	FC 70	67					LD	H, A
64625	FC 71	ED	5B	04	FF		LD	DE, (DEP3A)
64629	FC 75	A7					AND	A
64630	FC 76	ED	52				SBC	HL, DE
64632	FC 78	38	04				JR	C, PRSTP
64634	FC 7A	E1					POP	HL
64635	FC 7B	E1					POP	HL
64636	FC 7C	18	33				JR	PRMER
64638	FC 7E	CD	6B	0D		PRSTP	CALL	0D6BH
64641	FC 81	11	10	FF			LD	DE, MSSG
64644	FC 84	3E	03				LD	A, 3
64646	FC 86	CD	0E	FD			CALL	INPUT
64649	FC 89	28	0B				JR	Z, DEFLT
64651	FC 8B	AF					XOR	A
64652	FC 8C	BC					CP	H
64653	FC 8D	20	0A				JR	NZ, LDSTP
64655	FC 8F	BD					CP	L
64656	FC 90	20	07				JR	NZ, LDSTP
64658	FC 92	E1					POP	HL
64659	FC 93	E1					POP	HL
64660	FC 94	18	1B				JR	PRMER
64662	FC 96	21	0A	00		DEFLT	LD	HL, 10
64665	FC 99	22	06	FF		LDSTP	LD	(DEP4), HL
64668	FC 9C	D1					POP	DE
64669	FC 9D	CD	A9	30			CALL	30A9H
64672	FC A0	30	03				JR	NC, OVERF
64674	FC A2	E1					POP	HL
64675	FC A3	18	0C				JR	PRMER
64677	FC A5	ED	5B	04	FF	OVERF	LD	DE, (DEP3A)
64681	FC A9	19					ADD	HL, DE
64682	FC AA	D1					POP	DE
64683	FC AB	38	04				JR	C, PRMER

64685	FC	AD	ED	52	
64687	FC	AF	38	14	
64689	FC	B1	CD	6B	0D
64692	FC	B4	11	10	FF
64695	FC	B7	3E	04	
64697	FC	B9	CD	0A	0C
64700	FC	BC	21	3B	5C
64703	FC	BF	CB	9E	
64705	FC	C1	E1		
64706	FC	C2	C3	9C	12
64709	FC	C5	F3		
64710	FC	C6	CD	6B	0D
64713	FC	C9	DD	21	1F FE
64717	FC	CD	CD	6B	FD
64720	FC	D0	DD	21	53 FE
64724	FC	D4	CD	6B	FD
64727	FC	D7	2A	00	FF
64730	FC	DA	ED	5B	04 FF
64734	FC	DE	ED	4B	02 FF
64738	FC	E2	7C		
64739	FC	E3	B8		
64740	FC	E4	C2	EB	FC
64743	FC	E7	7D		
64744	FC	E8	B9		
64745	FC	E9	28	13	
64747	FC	EB	72		
64748	FC	EC	23		
64749	FC	ED	73		
64750	FC	EE	23		
64751	FC	EF	4E		
64752	FC	F0	23		
64753	FC	F1	46		
64754	FC	F2	09		
64755	FC	F3	23		
64756	FC	F4	EB		
64757	FC	F5	ED	4B	06 FF
64761	FC	F9	09		
64762	FC	FA	EB		
64763	FC	FB	C3	DE	FC
64766	FC	FE	DD	21	63 FE
64770	FD	02	CD	6B	FD
64773	FD	05	DD	21	6F FE
64777	FD	09	CD	6B	FD
64780	FD	0C	FB		
64781	FD	0D	C9		
64782	FD	0E	CD	0A	0C
64785	FD	11	21	00	00
64788	FD	14	06	04	
64790	FD	16	AF		
64791	FD	17	32	08	5C
64794	FD	1A	3A	08	5C
64797	FD	1D	D6	30	
64799	FD	1F	FE	0A	
64801	FD	21	30	01	
64803	FD	23	C5		
64804	FD	24	06	00	
64806	FD	26	4F		
64807	FD	27	54		
64808	FD	28	5D		
64809	FD	29	29		
64810	FD	2A	29		
64811	FD	2B	19		
64812	FD	2C	29		
64813	FD	2D	09		
64814	FD	2E	C1		
64815	FD	2F	C6	30	
64817	FD	31	D7		
64818	FD	32	10	E2	
64820	FD	34	FE	DD	
64822	FD	36	20	04	
64824	FD	38	78		
64825	FD	39	FE	04	
64827	FD	3B	C9		
64828	FD	3C	FE	DC	

	PRMER	SBC	HL, DE
		JR	C, DSBLI
		CALL	0D6BH
		LD	DE, MSSG
		LD	A, 4
		CALL	0C0AH
		LD	HL, FLAGS
		RES	3, (HL)
		POP	HL
		JP	129CH
	DSBLI	DI	
		CALL	0D6BH
	PROC1	LD	IX, CASE1
		CALL	MNSBR
	PROC2	LD	IX, CASE2
		CALL	MNSBR
	PROC3	LD	HL, (DEP1)
		LD	DE, (DEP3A)
	TSTND	LD	BC, (DEP2)
		LD	A, H
		CP	B
		JP	NZ, RSQNC
		CP	AC, L
		CP	
		JR	Z, PROC4
	RSQNC	LD	(HL), D
		INC	HL
		LD	(HL), E
		INC	HL
		LD	C, (HL)
		INC	HL
		LD	B, (HL)
		ADD	HL, BC
		INC	HL
	EX	DE, HL	
		LD	BC, (DEP4)
		ADD	HL, BC
		EX	DE, HL
		JP	TSTND
	PROC4	LD	IX, CASE3
		CALL	MNSBR
	PROC5	LD	IX, CASE4
		CALL	MNSBR
		EI	
		RET	
	INPUT	CALL	0C0AH
		LD	HL, 0
		LD	B, 4
	INIT	XOR	A
		LD	(LASTK), A
	MLTPL	LD	A, (LASTK)
		SUB	48
		CP	10
		JR	NC, ENTER
		PUSH	BC
		LD	B, 0
		LD	C, A
		LD	D, H
		LD	E, L
		ADD	HL, HL
		ADD	HL, HL
		ADD	HL, DE
		ADD	HL, HL
		ADD	HL, BC
		POP	BC
		ADD	A, 48
		RST	10H
		DJNZ	INIT
	ENTER	CP	-35
		JR	NZ, CURSL
		LD	A, B
		CP	4
		RET	
	CURSL	CP	-36

64830	FD 3E	28	0B	
64832	FD 40	78		
64833	FD 41	A7		
64834	FD 42	20	D6	
64836	FD 44	3A	08	5C
64839	FD 47	D6	30	
64841	FD 49	18	E9	
64843	FD 4B	78		
64844	FD 4C	FE	04	
64846	FD 4E	28	CA	
64848	FD 50	3E	08	
64850	FD 52	D7		
64851	FD 53	3E	20	
64853	FD 55	D7		
64854	FD 56	3E	08	
64856	FD 58	D7		
64857	FD 59	C5		
64858	FD 5A	11	FF	FF
64861	FD 5D	00	OA	OO
64864	FD 60	A7		
64865	FD 61	ED	42	
64867	FD 63	13		
64868	FD 64	30	FB	
64870	FD 66	EB		
64871	FD 67	C1		
64872	FD 68	04		
64873	FD 69	18	AB	
64875	FD 6B	2A	53	5C
64878	FD 6E	7E		
64879	FD 6F	FE	40	
64881	FD 71	D0		
64882	FD 72	22	08	FF
64885	FD 75	16	01	
64887	FD 77	1E	00	
64889	FD 79	D5		
64890	FD 7A	CD	88	19
64893	FD 7D	C2	11	FE
64896	FD 80	23		
64897	FD 81	7E		
64898	FD 82	EB		
64899	FD 83	21	0A	FF
64902	FD 86	01	06	00
64905	FD 89	ED	B1	
64907	FD 8B	EB		
64908	FD 8C	28	1F	
64910	FD 8E	FE	F8	
64912	FD 90	20	77	
64914	FD 92	23		
64915	FD 93	7E		
64916	FD 94	CD	B6	18
64919	FD 97	FE	22	
64921	FD 99	20	06	
64923	FD 9B	23		
64924	FD 9C	BE		
64925	FD 9D	20	FC	
64927	FD 9F	18	F1	
64929	FD A1	FE	0D	
64931	FD A3	28	6C	
64933	FD A5	FE	3A	
64935	FD A7	28	60	
64937	FD A9	FE	CA	
64939	FD AB	20	E5	
64941	FD AD	23		
64942	FD AE	7E		
64943	FD AF	FE	0D	
64945	FD B1	28	5E	
64947	FD B3	FE	3A	
64949	FD B5	28	52	
64951	FD B7	D6	30	
64953	FD B9	FE	0A	
64955	FD BB	DA	AD	FD
64958	FD BE	FE	DE	
64960	FD C0	20	0F	
64962	FD C2	01	06	00

DELET

DIVSN

MNSBR
PRGRM

SRCHS

SAVE

LINE

NEXTB

OTHER

CLCLT

JR	Z, BELET
LD	A, B
AND	A
JR	NZ, MLTPL
LD	A, (LASTK)
SUB	48
JR	ENTER
LD	A, B
CP	4
JR	NZ, MLTPL
LD	A, 8
RST	10H
LD	A, 20H
RST	10H
LD	A, 8
RST	10H
PUSH	BC
LD	DE, -1
LD	BC, 10
AND	A
SBC	HL, BC
INC	DE
JR	NC, DIVSN
EX	DE, HL
POP	BC
INC	B
JR	INIT
LD	HL, (PROG)
LD	A, (HL)
CP	64
RET	NC
LD	(LNADD), HL
LD	D, 1
LD	E, 0
PUSH	DE
CALL	1988H
JP	NZ, NEXTL
INC	HL
LD	A, (HL)
EX	DE, HL
LD	HL, TOKEN
LD	BC, 6
CPIR	
EX	DE, HL
JR	Z, CLCLT
CP	0F8H
JR	NZ, NEXTS
INC	HL
LD	A, (HL)
CALL	18B6H
CP	22H
JR	NZ, OTHER
INC	HL
CP	(HL)
JR	NZ, NEXTB
JR	LINE
CP	0DH
JR	Z, NEXTL
CP	3AH
JR	Z, NEXTS
CP	0CAH
JR	NZ, LINE
INC	HL
LD	A, (HL)
CP	0DH
JR	Z, NEXTL
CP	3AH
JR	Z, NEXTS
SUB	48
CP	10
JP	C, CLCLT
CP	-34
JR	NZ, EXPR
LD	BC, 6

64965	FD	C5	09		
64966	FD	C6	7E		
64967	FD	C7	FE	0D	
64969	FD	C9	CA	04	FE
64972	FD	CC	FE	3A	
64974	FD	CE	CA	04	FE
64977	FD	D1	DD	E5	
64979	FD	D3	E1		
64980	FD	D4	11	63	FE
64983	FD	D7	A7		
64984	FD	D8	ED	52	
64986	FD	DA	20	2D	
64988	FD	DC	3E	02	
64990	FD	DE	CD	01	16
64993	FD	E1	11	10	FF
64996	FD	E4	3E	06	
64998	FD	E6	FB		
64999	FD	E7	CD	0A	0C
65002	FD	EA	F3		
65003	FD	EB	2A	08	FF
65006	FD	EE	46		
65007	FD	EF	23		
65008	FD	F0	4E		
65009	FD	F1	CD	1B	1A
65012	FD	F4	3E	3A	
65014	FD	F6	D7		
65015	FD	F7	C1		
65016	FD	F8	C5		
65017	FD	F9	48		
65018	FD	FA	06	00	
65020	FD	FC	CD	1B	1A
65023	FD	FF	3E	0D	
65025	FE	01	D7		
65026	FE	02	18	05	
65028	FE	04	2B		
65029	FE	05	2B		
65030	FE	06	2B		
65031	FE	07	DD	E9	
65033	FE	09	2A	08	FF
65036	FE	0C	D1		
65037	FE	0D	14		
65038	FE	0E	C3	77	FD
65041	FE	11	D1		
65042	FE	12	2A	08	FF
65045	FE	15	23		
65046	FE	16	23		
65047	FE	17	4E		
65048	FE	18	23		
65049	FE	19	46		
65050	FE	1A	09		
65051	FE	1B	23		
65052	FE	1C	C3	6E	FD
65055	FE	1F	7E		
65056	FE	20	23		
65057	FE	21	66		L
65058	FE	22	6F		
65059	FE	23	CD	6E	19
65062	FE	26	CA	09	FE
65065	FE	29	AF		
65066	FE	2A	CD	01	16
65069	FE	2D	11	10	FF
65072	FE	30	3E	05	
65074	FE	32	CD	0A	0C
65077	FE	35	2A	08	FF
65080	FE	38	46		
65081	FE	39	23		
65082	FE	3A	4E		
65083	FE	3B	CD	1B	1A
65086	FE	3E	3E	3A	
65088	FE	40	D7		
65089	FE	41	C1		
65090	FE	42	48		
65091	FE	43	06	00	
65093	FE	45	CD	1B	1A

ADD	HL, BC	
	LD	A, (HL)
	CP	ODH
JP	Z, RTRN1	
	CP	3AH
	JP	Z, RTRN1
EXPR	PUSH	IX
	POP	HL
	LD	DE, CASE3
AND	A	
	SBC	HL, DE
	JR	NZ, NEXTS
	LD	A, 2
	CALL	1601H
	LD	DE, MSSG
	LD	A, 6
	EI	
	CALL	0CAH
	DI	
	LD	HL, (LNADD)
	LD	B, (HL)
	INC	HL
LD	C, (HL)	
	CALL	1A1BH
	LD	A, 3AH
	RST	10H
	POP	BC
	PUSH	BC
	LD	C, B
	LD	B, 0
	CALL	1A1BH
	LD	A, 0DH
	RST	10H
	JR	NEXTS
RTRN1	DEC	HL
	DEC	HL
	DEC	HL
	JP	(IX)
NEXTS	LD	HL, (LNADD)
	POP	DE
	INC	D
	JP	SRCHS
NEXTL	POP	DE
	LD	HL, (LNADD)
	INC	HL
	INC	HL
	LD	C, (HL)
	INC	HL
	LD	B, (HL)
	ADD	HL, BC
	INC	HL
	JP	PRGRM
CASE1	LD	A, (HL)
	INC	HL
	D	H, (HL)
	LD	L, A
	CALL	196EH
	JP	Z, NEXTS
ERROR	XOR	A
	CALL	1601H
	LD	DE, MSSG
	LD	A, 5
	CALL	0C0AH
	LD	HL, (LNADD)
	LD	B, (HL)
	INC	HL
	LD	C, (HL)
	CALL	1A1BH
	LD	A, 3AH
	RST	10H
	POP	BC
	LD	C, B
	LD	B, 0
	CALL	1A1BH

65096	FE	48	21	3B	5C
65099	FE	4B	CB	9E	
65101	FE	4D	E1		
65102	FE	4E	E1		
65103	FE	4F	FB		
65104	FE	50	C3	9C	12
65107	FE	53	E5		
65108	FE	54	7E		
65109	FE	55	23		
65110	FE	56	66		
65111	FE	57	6F		
65112	FE	58	CD	6E	19
65115	FE	5B	EB		
65116	FE	5C	E1		
65117	FE	5D	73		
65118	FE	5E	23		
65119	FE	5F	72		
65120	FE	60	C3	09	FE
65123	FE	63	5E		
65124	FE	64	23		
65125	FE	65	56		
65126	FE	66	1A		
65127	FE	67	77		
65128	FE	68	13		
65129	FE	69	2B		
65130	FE	6A	1A		
65131	FE	6B	77		
65132	FE	6C	C3	09	FE
65135	FE	6F	E5		
65136	FE	70	7E		
65137	FE	71	23		
65138	FE	72	66		
65130	FE	73	6F		
65140	FE	74	11	00	FF
65143	FE	77	01	18	FC
65146	FE	7A	CD	F4	FE
65149	FE	7D	12		
65150	FE	7E	01	9C	FF
65153	FE	81	CD	F4	FE
65156	FE	84	13		
65157	FE	85	12		
65158	FE	86	0E	F6	FE
65160	FE	88	CD	F4	FE
65163	FE	8B	13		
65164	FE	8C	12		
65165	FE	8D	7D		
65166	FE	8E	C6	30	
65168	FE	90	13		
65169	FE	91	12		
65170	FE	92	21	00	FF
65173	FE	95	01	04	00
65176	FE	98	3E	30	
65178	FE	9A	BE		
65179	FE	9B	C2	A3	FE
65182	FE	9E	23		
65183	FE	9F	0D		
65184	FE	A0	C3	9A	FE
65187	FE	A3	E3		
65188	FE	A4	11	FC	FF
65191	FE	A7	19		
65192	FE	A8	16	00	
65194	FE	AA	2B		
65195	FE	AB	14		
65196	FE	AC	7E		
65197	FE	AD	D6	30	
65199	FE	AF	FE	0A	FE
65201	FE	B1	DA	AA	FE
65204	FE	B4	23		
65205	FE	B5	C5		
65206	FE	B6	79		
65207	FE	B7	92		
65208	FE	B8	32	04	FF
65211	FE	BB	D2	CC	FE
65214	FE	BE	ED	44	

CASE2

CASE3

CASE4

NEWAS

RTRN2

OLDAS

SHIFT

RECLM

LD	HL, FLAGS
RES	3, (HL)
POP	HL
POP	HL
EI	
JP	129CH
PUSH	HL
LD	A, (HL)
INC	HL
LD	H, (HL)
LD	L, A
CALL	196EH
EX	DE, HL
POP	HL
LD	(HL), E
INC	HL
LD	(HL), D
JP	NEXTS
LD	E, (HL)
INC	HL
LD	D, (HL)
LD	A, (DE)
LD	(HL), A
INC	DE
DEC	HL
LD	A, (DE)
LD	(HL), A
JP	NEXTS
PUSH	HL
LD	A, (HL)
INC	HL
LD	H, (HL)
LD	L, A
LD	DE, DEP1
LD	BC, -1000
CALL	CONV
LD	(DE), A
LD	BC, -100
CALL	CONV
INC	DE
LD	(DE), A
LD	C, -10
CALL	CONV
INC	DE
LD	(DE), A
LD	A, L
ADD	A, 48
INC	DE
LD	(DE), A
LD	HL, DEP1
LD	BC, 4
LD	A, 48
CP	(HL)
JP	NZ, RTRN2
INC	HL
DEC	C
JP	NEWAS
EX	(SP), HL
LD	DE, -4
ADD	HL, DE
LD	D, 0
DEC	HL
INC	D
LD	A, (HL)
SUB	48
CP	10
JP	C, OLDAS
INC	HL
PUSH	BC
LD	A, C
SUB	D
LD	(DEP3A), A
JP	NC, INSRT
NEG	

65216	FE	C0	4F				
65217	FE	C1	3E	FF			
65219	FE	C3	32	05	FF		
65222	FE	C6	CD	E8	19		
65225	FE	C9	C3	D5	FE		
65228	FE	CC	4F				
65229	FE	CD	AF				
65230	FE	CE	32	05	FF		
65233	FE	D1	CD	55	16		
65236	FE	D4	23				
65237	FE	D5	EB				
65238	FE	D6	C1				
65239	FE	D7	E1				
65240	FE	D8	ED	B0			
65242	FE	DA	2A	08	FF		
65245	FE	DD	23				
65246	FE	DE	23				
65247	FE	DF	7E				
65248	FE	E0	23				
65249	FE	E1	66				
65250	FE	E2	6F				
65251	FE	E3	ED	4B	04	FF	
65255	FE	E7	09				
65256	FE	E8	EB				
65257	FE	E9	2A	08	FF		
65260	FE	EC	23				
65261	FE	ED	23				
65262	FE	EE	73				
65263	FE	EF	23				
65264	FE	FP	72				
65265	FE	F1	C3	09	FE		
65268	FE	F4	3E	FF			
65270	FE	F6	09				
65271	FE	F7	3C				
65272	FE	F8	DA	F6	FE		
65275	FE	FB	ED	42			
65277	FE	FD	C6	30			
65279	FE	FF	C9				
65280	FF	00	00	00			
65282	FF	02	00	00			
65284	FF	04	00				
65285	FF	05	00				
65286	FF	06	00	00			
65288	FF	08	00	00			
65290	FF	0A	EC				
65291	FF	0B	ED				
65292	FF	0C	E5				
65293	FF	0D	F7				
65294	FF	0E	E1				
65295	FF	0F	F0				
65296	FF	10	80				
65297	FF	11	46	72	6F	6D	
65301	FF	15	A0				
65302	FF	16	54	6F			
65304	FF	18	A0				
65305	FF	19	53	74	61	72	74
65310	FF	1E	A0				
65311	FF	1F	53	74	65	70	
65315	FF	23	A0				
65316	FF	24	50	61	72	61	6D
			65	74	65	72	
65325	FF	2D	20	45	72	72	6F
65330	FF	32	F2				
65331	FF	33	49	6D	70	6F	73
			73	69	62	6C	65
65341	FF	3D	20	52	65	6E	75
			6D	62	65	72	69
			6E	67			
65353	FF	49	A0				
65354	FF	4A	45	78	70	72	65
			73	73	69	6F	6E
65364	FF	54	20	49	6E		
65367	FF	57	A0				

	LD	C, A
	LD	A, -1
	LD	(DEP3B), A
	CALL	19E8H
INSRT	JP	LDASC
	LD	C, A
	XOR	A
	LD	(DEP3B), A
	CALL	1655H
LDASC	INC	HL
	EX	DE, HL
	POP	BC
	POP	HL
	LDIR	
LINK	LD	HL, (LNADD)
	INC	HL
	INC	HL
	LD	A, (HL)
	INC	HL
	LD	H, (HL)
	LD	L, A
	LD	BC, (DEP3A)
	ADD	HL, BC
	EX	DE, HL
	LD	HL, (LNADD)
	INC	HL
	INC	HL
LD	INC	(HL), E
	INC	HL
	LD	(HL), D
	JP	NEXTS
CONV	LD	A, -1
SUBTR	ADD	HL, BC
	INC	A
	JP	C, SUBTR
	SBC	HL, BC
	ADD	A, 48
	RET	
DEP1	DEFS	2
DEP2	DEFS	2
DEP3A	DEFS	1
DEP3B	DEFS	1
DEP4	DEFS	2
LNADD	DEFS	2
TOKEN	DEFB	0ECH
	DEFB	0EDH
	DEFB	0E5H
	DEFB	0F7H
	DEFB	0E1H
	DEFB	0F0H
MSSG	DEFB	80H
	DEFM	"From"
	DEFB	0A0H
	DEFB	"To"
	DEFB	0A0H
	DEFM	"Start"
	DEFB	0A0H
	DEFM	"Step"
	DEFB	0A0H
	DEFM	"Parameter"
	FEFM	"Error"
	DEFB	0F2H
	DEFM	"Impossible"
	DEFM	"Renumbering"
	DEFB	0A0H
	DEFM	"Expression"
	DEFM	"In"
	DEFB	0A0H

END

Recensioni Libri

di **Lucio Bragagnolo**

Alla scoperta del BASIC Spectrum

di **Mike Lord**
ed. **Gruppo Editoriale Jackson**

Il primo protagonista di questa puntata di PRINT è dedicato ai possessori di Spectrum, siano dilettanti o esperti programmatori, che potranno trovarvi suggerimenti, consigli e programmi preziosi per il loro hobby o lavoro. Accompagnato da una cassetta contenente tutti i programmi in esso presenti (secondo una piacevolissima tradizione instaurata dalla Jackson da qualche tempo a questa parte), il libro funziona sia come strumento di apprendimento della programmazione che come raccolta di routine e programmi utili, piccoli come dimensioni, ma numerosissimi (circa cinquanta).

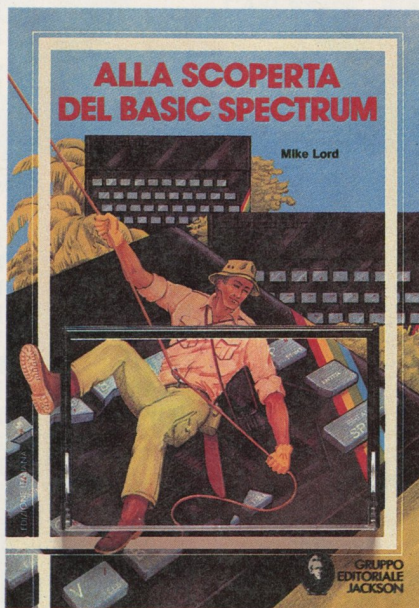
Una considerazione generale da fare circa i listati è la loro funzione prevalentemente dimostrativa: sebbene infatti siano presenti alcuni programmi direttamente fruibili (come "Labirinto" e altri, di cui diremo più avanti) gli altri sono veramente ridotti all'osso, senza abbellimenti grafici particolari o un output accurato. Servono, in definitiva, perché il lettore li capisca e li usi successivamente per le sue particolari applicazioni. I listati sono ottenuti con una Seikosha GP50S, ma sono perfettamente leggibili (e il più delle volte di lunghezza più che contenuta).

Vendendo al libro vero e proprio, esso è diviso in quattordici capitoli e tre utili appendici, queste ultime riguardanti le locazioni di memoria più interessanti dello Spectrum (variabile di sistema, schermo, attributi, printer buffer), notizie e consigli sulla velocizzazione dei programmi (sezione insolita e preziosissima per i perfezionisti) e un confronto con i vari dialetti BASIC esistenti, di aiuto per chi sogna conversioni di programmi scritti su altri calcolatori (eccettuate ovviamente le diverse POKE e PEEK!).

Quanto ai vari capitoli, si tratta di una buona descrizione del BASIC, di cosa è un calcolatore e delle peculiarità dello

**I LIBRI
vengono
offerti con il 10%
di sconto sul
prezzo di copertina**

Spectrum, a partire dal concetto generico di memoria fino ad arrivare alla descrizione dei bug presenti nella ROM dello ZX. Per credere, provare questi



esempietti (di cui uno è diligentemente riportato su cassetta):

```
10 PRINT AT 0,0; "a": REM (sul libro
- e sulla cassetta -, è riportata una
virgola anziché un punto e virgola;
si tratta di un piccolo errore)
20 PRINT SCREEN$(0,0)
30 PRINT "b"+SCREEN$(0,0)
40 PRINT CODE SCREEN$(0,0)
50 PRINT 2+CODE SCREEN$(0,0)
60 IF CODE "a"= CODE SCREEN$(
(0,0) THEN PRINT "ok"
```

oppure

```
PRINT -65534-2
PRINT INT -65536 (già noto)
FOR a=-65530 TO -65540 STEP
-1: PRINT a: NEXT a
PRINT "A" + STR$(0.1)+"B"
```

Il resto è "normale" fino al capitolo 7, dedicato ai numeri casuali e alla poesia casuale. Gli altri capitoli meritevoli di citazione sono il 9 ("Belle arti"), godibile graficamente, contenente anche un programma per ridefinire completamente il set di caratteri in uso; il 10 ("Suoni e colori"), in cui vengono spiegati in dettaglio, tra le altre cose, gli attributi del video e l'uso di tutti i caratteri di controllo sia nei programmi che nei listati (argomento questo ignoto a un grandissimo numero di sinclairisti); il 12, "GO SUB, DEF FN e bello stile", in cui si spazia dalle tematiche della programmazione strutturata all'uso di DEF FN - l'istruzione trascurata per eccellenza - a 2 ottimi programmi, "3D" e "Labirinto"; il 13 (medie, istogrammi, sistemi di equazione, piccoli database) e, infine, il 14, intitolato "Programmi di utilità, stranezze e routine ricorrenti", dove il lettore imparerà, per esempio, come inserire una linea 0 nei propri listati o come controllare via software l'integrità di ROM e RAM; in omaggio (si fa per dire) due programmini, rispettivamente un renumber e un convertitore di base numerica, che traduce, poniamo, un numero scritto in notazione binaria nell'equivalente decimale, esadecimale, ottale o a proprio piacere.

Il linguaggio usato è chiaro e scorrevole, e i listati sono anche qui ottenuti con una Seikosha ben oliata ed efficiente. Qua e là si vedono anche alcune copie del video (sempre ottenute con la stampante), utili per farsi un'idea dell'output dei programmi interessati.

Per concludere, la cassetta: provatela, ha funzionato senza errori dal primo all'ultimo caricamento (a qualcuno sembrerà ovvio, ma l'esperienza in questo campo insegna a non dare mai certe cose per scontate). I programmi ci sono tutti, e funzionano: le eccezioni in questo senso vengono dal programmino che abbiamo riportato più sopra (dove una virgola ha sostituito il punto e virgola, che avrebbe dovuto esserci) e da un paio di altri programmi tecnicamente perfetti, ma che presentano lievi difetti di output dovuti alla traduzione italiana di termini inglesi equivalenti, ma di diversa lunghezza (vedere "bioritmi": molto bello da vedere, ma la scritta "intellettuale" viene spezzata proprio in corrispondenza dell'ultima "e"; in più "wks", evidente abbreviazione di "weeks" (settimane), non è apparsa altrettanto evidente al traduttore, che l'ha perfino storpiata in "vks"). Questi sono comunque peccati veniali, che, ripetiamo, non inficiano il perfetto funzionamento dei programmi e coste-

**I libri possono essere
ordinati utilizzando
il coupon
di pag. 98**

ranno non più di cinque minuti di modifiche anche al meno esperto dei programmatori. Il programma più bello è senz'altro "Labirinto": giocabilità a parte, è eccezionale la fase di generazione del labirinto, dove si può vedere lo Spectrum "pensare" al cammino da seguire dalla prima casella fino all'ultima: l'effetto è decisamente affascinante — specie definendo un labirinto delle massime dimensioni consentite — e anche parlando di pura e semplice programmazione la routine impiegata è tutto sommato ineccepibile.

In definitiva, assegnamo un buon voto anche a "Alla scoperta del BASIC Spectrum", libro che contribuisce ad alzare il tasso di qualità medio (già di per sé notevole) delle edizioni Jackson.

Il BASIC in 30 ore per lo ZX81

di Clive Prigmore
ed. Gruppo Editoriale Jackson

Ai più accorti tra i lettori abituali di PRINT non sarà mancata, leggendo il titolo di questa recensione, un'impressione di "dejà vu", di già sentito. L'impressione è esatta: il libro in esame differisce dal suo predecessore "Il BASIC in 30 ore per lo Spectrum" solo per il computer oggetto della trattazione, il veterano, ma non certo pensionato, ZX81: e che i suoi possessori possano stare tranquilli lo indica la notizia (nel momento in cui scriviamo recentissima) diffusa della Sinclair, secondo la quale, mantenendosi la vendita dello ZX81 su livelli soddisfacenti, l'azienda non ha alcuna intenzione di cessarne la produzione.

In effetti lo ZX81 sta diventando ciò che in campo automobilistico è rappresentato dalla 500: qualcosa di vecchio, ma ancora funzionale, utile sia all'utente smalzato che a chi vuole cominciare spendendo pochissimo (ormai uno ZX81 costa meno di certe calcolatrici tascabili...): soprattutto a questi ultimi si rivolge "Il BASIC in 30 ore per lo ZX81", testo notevolissimo per chiarezza di impostazione e facilità d'uso. La struttura del libro è perfettamente identica a quella del suo predecessore dedicato allo Spectrum, fino ad avere identici perfino alcuni capitoli. Anche questa recensione, per forza di cose, ricalcherà quindi quella già scritta a suo tempo.

Il segreto, se così si può chiamare, dell'opera consiste nell'ennesimo uovo di Colombo: scrivere un libro che insegni a programmare... scrivendo un libro che insegni realmente a programmare. Cosa vuol dire? Cerchiamo di spiegarci meglio. Ciò che intendiamo sottolineare è che finalmente siamo di fronte a un sussidio didattico vero e proprio, scritto, impaginato, disegnato, progettato a scopo di insegnamento, dove prima la maggioranza dei libri puntava so-

prattutto a dare il massimo di informazione nel minimo di pagine, col risultato di sconcertare il lettore, letteralmente sorpassato, dopo i primi capitoli introduttivi, dal ritmo frenetico del libro.

Beninteso, non si vuol negare l'utilità di tutte le pubblicazioni apparse finora; vogliamo però sottolineare che per un neofita l'approccio più produttivo è quello usato in "Il BASIC..".

Sfogliandolo distrattamente, l'aspetto è simile a quello di un libro di testo per le scuole: periodi evidenziati in verde, numerosissime illustrazioni chiarificatrici, piccoli listati presenti dappertutto (anche qui le linee di programma significative sono visualizzate in verde), con una differenza: la mancanza di testi lunghissimi e verbosi. La sintesi prevalente sulla logorrea, in omaggio al vecchio detto inglese secondo cui chi impiega sei parole potendone usare anche solo cinque è capace di qualsiasi delitto. Colpisce anche la spaziatura tra le varie informazioni, non riempitivo per mancanza di idee quanto aiuto alla chiarezza: chi legge deve avere il tem-



po di "digerire" lo scritto e meditarci sopra.

Approfondendo l'esame, possiamo ora vedere quali sono le direttrici secondo le quali l'autore ha impostato il libro. Come leggiamo dall'introduzione, per saper utilizzare al meglio un personal computer bisogna conoscere per prima cosa il suo dialetto BASIC e, in seguito, imparare a realizzare buone strutture di programma. Il resto (ad esempio la tastiera) viene dopo, e difatti la gran parte del libro si occupa dei primi due punti. Altra cosa: il corso può essere seguito anche senza possedere un computer! Vi sono infatti esempi ed esercizi riservati agli utenti ZX81, ed altri accessibili a chiunque. Infine, non si parla di hardware: tutto ciò che bisogna sapere per programmare un calcolatore è la maniera migliore per fornirgli un problema, così che esso risponda come desiderato: il resto, almeno finché si

impara, non ha eccessiva importanza. Aggiungiamo a questo quadro di insieme la presenza di numerosi test, esercizi, verifiche, e sarà già possibile a questo punto farsi un'idea abbastanza precisa dell'insieme. Scendendo ancora maggiormente nei particolari, vediamo quali sono gli argomenti esaminati. È importante sottolineare come nel libro non venga trattato l'intero BASIC ZX81: ci sono istruzioni (PLOT, POKE, USR...) e argomenti (variabili di sistema, linguaggio macchina, routine della ROM...) che non sono di utilità per chi impara a programmare, e che anzi possono condurre a una pericolosa confusione mentale e quindi non sono trattati.

I capitoli del libro sono 8: il primo di essi è dedicato all'analisi di un elementare programma di somma di due numeri, in cui vengono introdotti LIST, SAVE e LOAD. Il capitolo 2 si intitola "Prendere le decisioni", e in esso si parla di IF...THEN, della logica booleana (illustrata molto chiaramente), di algoritmi, flowchart e dell'uso di GO TO, su cui l'autore si allinea con l'opinione dei più celebri santoni dell'informatica, sconsigliandone l'uso se non quando sia strettamente necessario. Si prosegue con stringhe e liste, argomenti che consentono la trattazione di temi come variabili, separatori, file, FOR...NEXT, ordinamento dell'output sullo schermo e altro ancora. Non manca neppure una parte dedicata allo slicing, completa della traduzione degli operatori RIGHT\$, MID\$ e LEFT\$, propri del BASIC Microsoft; si parla anche di sicure negli input, argomento che (esperienza redazionale) probabilmente pochissimi possono permettersi di ignorare.

Il capitolo 6 ("Soprattutto sui dadi e giochi") tratta di casualità (RND) e operazioni su stringhe, mentre nel capitolo 7 ("Elaborazione numerica") l'obiettivo è puntato su temi come il calcolo delle medie, la rappresentazione numerica, l'uso di INT e ABS, il calcolo iterativo (ad esempio per trovare una radice quadrata o numeri primi sempre più grandi) e il concetto, solitamente sconosciuto o comunque ignorato, di "dry run", vale a dire la simulazione di funzionamento del programma stesso, a scopo di perfezionamento e debugging. Il capitolo 8, infine, è una "Introduzione all'elaborazione dati". Breve, ma succoso, condensa in poche pagine nozioni utilissime sui vari sistemi di ricerca e ordinamento di un file, su come stendere un menu di programma e come organizzare un output sotto forma di liste o tabelle.

Come già per il testo analogo dedicato allo Spectrum, il nostro giudizio su "Il BASIC in 30 ore per lo ZX81" è largamente positivo: non è sempre facile trovare, nell'inflazionatissimo panorama biblioinformatico, testi come questo, realmente utili e orientati verso le effettive esigenze dei lettori.

ZX CLUB

**I club
Sinclair**

Nascita di alcuni club, constatazione della piena attività di altri già fiorenti, assestamento e consolidamento di altri ancora, fondazione dei primi club di utenti QL... il panorama del settore appare tutt'altro che stanco e smobilitante come le cassandre dei "computer-usa-e-getta" vorrebbero farci credere. La nostra risposta è ancora quella di dedicare, nei limiti delle nostre possibilità, uno spazio per tutti, per la soddisfazione di chi conduce un'attività amatoriale, ma spesso a livelli professionali, e la curiosità o l'interesse di chi magari non sapeva del club Sinclair proprio dietro l'angolo, e vorrebbe trovare qualcuno con cui condividere la passione per l'informatica e, più concretamente, dare e ricevere programmi, consigli, libri, assistenza. A voi ZX CLUB!

Gli annunci di ZX Club

QUANTUM LEAP USER CLUB cerca nuovi soci in tutta Italia. Iscrizione gratuita, abbiamo già a disposizione software, libri e tanti consigli. Informazioni presso
Roberto GHEZZI
Via Volontari del sangue, 202
20099 SESTO SAN GIOVANNI (MI)
Tel. 02/2485511

Cerco ragazzi in Rapallo o in provincia di Genova possessori di ZX Spectrum per fondare club con scopo scambio di idee e software. Telefonare dalle 13 alle 15 a
Massimiliano GRASSI
Via Zunino, 19/1 - 16035 RAPALLO (GE)
Tel. 0185/272447

Iscriviti al **SINCLAIR NEW CLUB**. Con sole Lit. 6000 annue riceverai 5 programmi in omaggio a tua scelta, adesivi, sconti hard-soft; potrai inoltre usufruire della softeca e della biblioteca del club e approfittare di mille occasioni!
Gianpaolo GENTILI
Via Turati, 10 - 10024 MONCALIERI (TO)
Tel. 011/6407195

Il **SINCLAIR CLUB ROMA-TALENTI** cerca soci in tutta Italia sia tra coloro che vogliono usare lo Spectrum come videogioco sofisticato sia tra chi ricerca invece qualcosa di più. Per maggiori informazioni o per l'iscrizione rivolgersi a
Massimo D'ASCENZO
Via F. D'Ovidio, 109 - 00137 ROMA
Tel. 06/8280043

Come capoclub annuncio felice la nascita del **BUG-BUSTERS ZX Club**, giovane come creazione ma già effervescente e vulcanico nelle iniziative, tra le quali segnalo il nostro mensile-bollettino con articoli sullo Spectrum, trucchi, POKE e mappe dei migliori giochi, routine utili e molte altre cose che non elenco per non monopolizzare lo spazio di ZX Club.
Chi fosse interessato può contattarci a questo indirizzo:

BUG-BUSTERS ZX CLUB
c/o Andrea CICOGLIA
Via S. Quasimodo, 6/c
46023 GONZAGA (MN)
Tel. 0376/588555

A tutti i migliori auguri di **SUPERSINC** e, ovviamente, l'invito a collaborare a **ZX CLUB** con ogni mezzo: programmi, articoli, bollettini, trucchi e bug, adesivi, e chi più ne ha più ne metta!

Le notizie di ZX Club

Ricordate il **FIRENZE SINCLUB**, quello di "Il mago Numerello"? Ora ha cambiato sede: il nuovo indirizzo è **FIRENZE SINCLUB**
Via Elbano Gasperi, 33
50131 FIRENZE (FI)

Per maggiori informazioni vi consigliamo però di scrivere, per il momento, alla segreteria del club: l'indirizzo di questa è
Firenze Sinclub - c/o Marco Agresti -
Via Castelfidardo 6 - 50137 Firenze
Il cambio di indirizzo non ha però bloccato l'attività del club, di cui sentiremo riparlar presto...

I programmi di ZX Club

Il programma di questa puntata di **ZX CLUB "boxe"**, di Gianpaolo Gentili del Sinclair New Club di Moncalieri (To), di cui trovate più sopra l'indirizzo. Simpatico seppure semplice, ha un piccolo difetto: utilizzando la funzione **INKEY\$** i due pugili (manovrati ognuno da un giocatore) non possono muoversi contemporaneamente. Chi ha voglia di cimentarsi nella non difficile impresa potrà sostituire le **INKEY\$** con la funzione **IN**, così da consentire il movimento simultaneo di entrambi i boxeur. Buon divertimento!

Boxe

versione per **ZX Spectrum 16K/48K**

```
1 CLS
10 GO SUB 471: GO TO 32
20 IF p=o+1 THEN GO TO 431
21 IF p-1=o THEN GO TO 451
31 RETURN
35 FOR n=1 TO (30-4)/2: PRINT AT 11
,n; INVERSE 1;"BOXE!": PRINT
AT 11,n-1;" "
```

```
37 BEEP .05,n
39 NEXT n
41 FOR p=29 TO 14 STEP -1
43 BEEP .008,1: PRINT AT 10,p;"JI "
;AT 11,p-1;"RLK ";AT 12,p;"NM ";
AT 13,p;"PO "
44 BEEP .009,n
45 NEXT p
47 CLS : PRINT PAPER 3; INK 7;AT 1
,10;"ISTRUZIONI"
49 FOR n=1 TO 31
```

```

50 PRINT AT 2,n;" Il gioco consiste
   nel colpire l'avversario con
   il pugno in modo da mandarlo
   al tappeto. Quando lo si sara
   ' colpito per cinque volte la p
   artita si concludera'."
   NEXT n
51 PRINT FLASH 1; PAPER 7; INK 2;
   AT 21,1;"PREMI UN TASTO PER CONT
   INUARE"
61 PAUSE 0
71 PAPER 6: BORDER 6: INK 0:
   CLS
75 PRINT PAPER 2; INK 7;AT 1,10;"C
   OMANDI:"
76 PRINT AT 5,5;" Q<-->W";AT 7,7;"
   Q1 "
77 PRINT AT 5,20;" 9<-->0";AT 7,23;
   " 8R"
81 INPUT "INSERISCI IL NOME DEL GIO
   CATORE 1";a$
91 INPUT "INSERISCI IL NOME DEL GIO
   CATORE 2";b$
100 CLS
101 IF a$="" THEN LET a$="GIOCATORE
   -1"
111 IF b$="" THEN LET b$="GIOCATORE
   -2"
121 PRINT AT 0,0; INVERSE 1;a$;AT 0,
   20; INVERSE 1;b$
131 PRINT AT 1,0;"PUNT.1= ";AT 1,20;
   "PUNT.2= "
141 LET a=0: LET b=0
151 PRINT AT 1,8;a;AT 1,28;b
161 INK 5
171 PRINT AT 10,9;"{SG5}": PRINT
   AT 10,24;"{G5}";AT 11,9;"{SG5}":
   PRINT AT 11,24;"{G5}";AT 12,9;"
   {SG5}": PRINT AT 12,24;"{G5}"
181 INK 0
191 PRINT INK 5;AT 13,9;"{16SG8}";
   INK 1;AT 14,9;"{16SG8}";AT 15,9
   ;"{16SG8}"
195 FOR t=4 TO 1 STEP -1
197 PAUSE 55: PRINT INVERSE 1;AT 11
   ,16;t-1
199 NEXT t: PRINT AT 11,16;" "
201 FOR z=30 TO 50: BEEP .01,z:
   NEXT z
211 LET o=12: LET p=21
221 IF a=5 THEN PRINT INVERSE 1;
   FLASH 1;AT 5,5;"VINCE ";a$;
   AT 20,2;"premi un tasto per gioc
   are": PAUSE 0: CLS : GO TO 71
231 IF b=5 THEN PRINT INVERSE 1;
   FLASH 1;AT 5,5;"VINCE ";b$;
   AT 20,2;"premi un tasto per gioc
   are": PAUSE 0: CLS : GO TO 71
241 PRINT AT 9,o-2;" ";AT 9,o-1;" ";
   AT 10,o-2;" ";AT 10,o-1;" ";AT 1
   1,o-2;" ";AT 11,o-1;" ";AT 12,o-
   2;" ";AT 12,o-1;" ";AT 9,O-1;"A"
   : PRINT AT 9,O;"B"
251 PRINT AT 10,O-1;"C";AT 10,O;"D"
261 PRINT AT 11,O-1;"E";AT 11,O;"F"
271 PRINT AT 12,O-1;"G";AT 12,O;"H"

```

```

281 PRINT AT 9,p+1;" ";AT 9,p+2;" ";
   AT 10,p+1;" ";AT 10,p+2;" ";AT 1
   1,p+1;" ";AT 11,p+2;" ";AT 12,p+
   1;" ";AT 12,p+2;" ";AT 9,P+1;"I"
   ;AT 9,P;"J"
291 PRINT INK 2;AT 10,P+1;"K";AT 10
   ,P;"L"
301 PRINT AT 11,p+1;"M";AT 11,p;"N"
311 PRINT AT 12,P+1;"O";AT 12,P;"P"
321 PAUSE 0
331 IF INKEY$="w" THEN LET o=o+1
341 IF INKEY$="q" THEN LET o=o-1:
   PRINT AT 9,o+1;" ";AT 10,o+1;"
   ";AT 11,o+1;" ";AT 12,o+1;" "
351 IF INKEY$="9" THEN LET p=p-1
361 IF INKEY$="0" THEN LET p=p+1:
   PRINT AT 9,p-1;" ";AT 10,p-1;"
   ";AT 11,p-1;" ";AT 12,p-1;" "
371 IF INKEY$="1" THEN PRINT AT 10,
   o+1;"Q": BEEP .05,1: PRINT AT 10
   ,o+1;" ": GO SUB 20
381 IF INKEY$="8" THEN PRINT INK 2
   ;AT 10,p-1;"R": BEEP .05,.05:
   PRINT AT 10,p-1;" ": GO SUB 21
391 IF o<12 THEN LET o=12
401 IF P<13 OR p>21 THEN LET p=21:
   PRINT AT 9,o+1;" ";AT 10,o+1;"
   ";AT 11,o+1;" ";AT 12,o+1;"
   "
411 IF o+1>p THEN LET o=o-1
421 GO TO 221
431 PRINT AT 9,p;" ";AT 10,p;" ";
   AT 11,p;" ";AT 12,p;"K.O":
   BEEP .25,p: LET a=a+1
441 PRINT AT 1,8;a: GO TO 221
451 PRINT AT 9,o-1;" ";AT 10,o-1;"
   ";AT 11,o-1;" ";AT 12,o-2;"K.O
   .": BEEP .25,o: LET b=b+1
461 PRINT AT 1,28;b: GO TO 221
471 RESTORE 481: FOR i=USR "a" TO
   USR "r"+7: READ a: POKE i,a:
   NEXT i
481 DATA 0,0,1,3,7,7,3,1
491 DATA 0,0,128,192,64,240,192,0
501 DATA 1,15,20,20,20,23,16,31
511 DATA 0,192,64,64,92,126,30,254
521 DATA 15,8,15,8,15,3,3,3
531 DATA 192,64,192,64,192,128,128,1
   28
541 DATA 3,3,3,3,3,3,7,7
551 DATA 128,128,128,128,128,128,240
   ,240
561 DATA 0,0,128,192,224,224,192,128
571 DATA 0,0,1,3,2,15,3,0
581 DATA 128,248,40,40,40,232,8,248
591 DATA 0,3,2,2,58,126,120,127
601 DATA 240,240,240,240,240,192,192
   ,192
611 DATA 3,3,3,3,3,1,1,1
621 DATA 192,192,192,192,192,192,224
   ,224
631 DATA 1,1,1,1,1,1,15,15
641 DATA 0,0,0,14,15,255,255,14
651 DATA 0,0,0,112,240,255,255,112
661 RETURN

```


FINALMENTE!

La Softrivista che ti gasa!



**QUALCOSA DI SUPER, DI INEDITO,
DI IRRESISTIBILE**

**IL VERO GIOCO
COMINCIA ADESSO**

**IN EDICOLA
JACKSON SOFT
SERIE ORO**

I giochi esclusivi per Commodore 64 e Spectrum 48 K importati dall'Inghilterra, mai presentati in Italia. Una sfida Jackson al già visto, al già fatto, al... già registrato.



La prima puntata del fantastico, inedito **PYJAMARAMA**

Corri in edicola, il vero gioco comincia solo adesso e se sei davvero bravo partecipa alla "sfida al campione", utilizzando il tagliando che troverai sull'ultima pagina di copertina di ogni numero.



GRUPPO EDITORIALE JACKSON

SUPERSINC

CEDOLA DI ORDINAZIONE - CASSETTE
da compilare e spedire in busta chiusa a
J.soff - Viale Restelli, 5 - 20124 Milano - Tel. 6898228

LIBRI

ALLA SCOPERTA DEL BASIC
SPECTRUM
IL BASIC IN 30 ORE PER
LO ZX81

cod. AJAN034 L. 30.600
cod. AJAK035 L. 27.900

Ordino i seguenti libri per un importo totale di L. + L. 2.000
come contributo fisso per spese di spedizione

Cod. Cod.

Cod. Cod.

Contanti allegati Assegno allegato n°

Ho spedito l'importo a mezzo vaglia postale

Ho versato l'importo sul ccp n° 19445204 intestato a J.soff - Milano

Pagherò in contassegno al postino al ricevimento dei volumi (valido solo per i soci in Italia)

Nome

Cognome

Via

Città

Se richiesta fattura:
Cod. F. e P. Iva

Data

Firma

Per i residenti all'estero - pagamento anticipato (vaglia o versamento su ns. ccp)

SUPERSINC INPUT/OUTPUT

La rubrica INPUT/OUTPUT è gratuita ed aperta a tutti i lettori. Chi desidera comprare, vendere o cambiare hardware o software può inviare il tagliando a J.soff - Viale Restelli, 5 - 20124 MILANO

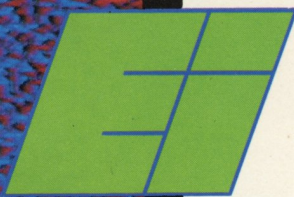
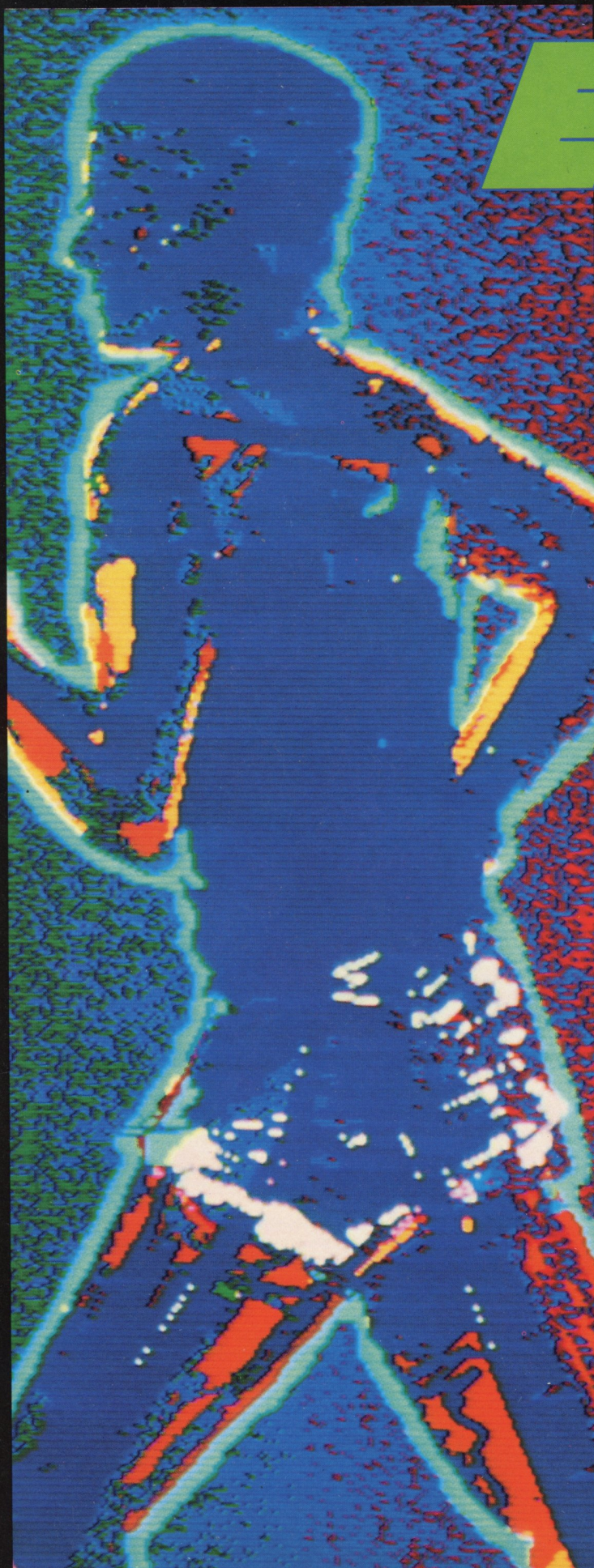
- COMPRO VENDO CAMBIO
 ZX80 ZX81 ZX Spectrum QL SOFTWARE
 PERIF.

Nome Cognome
Via C.A.P.
Città Tel.

SUPERSINC è bello, però... (ovvero suggerimenti, idee, critiche, richieste e tutto ciò che vi passa per la testa).

Nome Cognome
Via C.A.P.
Città Tel.

NON FARTI SUPERARE DAL PROGRESSO



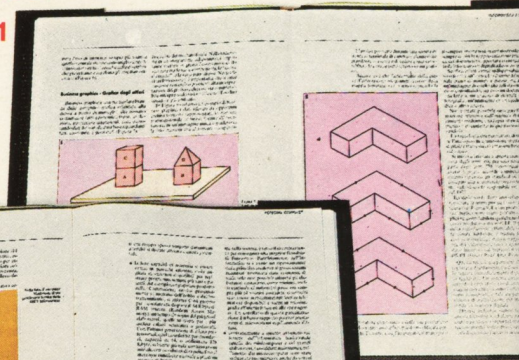
Aggiornati con gli "Aggiornamenti"

ENCICLOPEDIA DI ELETTRONICA & INFORMATICA

nuovidea

20 FASCICOLI SETTIMANALI DA RILEGARE
IN DUE NUOVI E SPLENDIDI VOLUMI

1



2

PER TE, PER IL TUO LAVORO, PER I TUOI STUDI...

1 **Aggiornamenti**

le nuove conquiste dell'Elettronica di Base, delle Comunicazioni, dell'Elettronica Digitale, dei Microprocessori, dell'Informatica...

2 **Il personal computer**

tutto quello che c'è da sapere sul Personal Computer: che cos'è e cosa fa; come fa e come si fa; i linguaggi di programmazione; le applicazioni...

E.I. si aggiorna e ti aggiorna

il 18 settembre

batti sul tempo il progresso!
Corri in edicola a comprare il primo fascicolo.

Solo L. 2500



**ANCORA UNA VOLTA
"PRIMI SUL FUTURO"**



**GRUPPO
EDITORIALE
JACKSON**

SAN FRANCISCO - LONDRA - MILANO